Analysis and Implementation of Security Algorithm for Healthcare Application

E. S. Selva Priya, Amritha K., Logeshwari C. and Aishwarya S.
Department of ECE, KCG College of Technology, KCG Nagar, Rajiv Gandhi Salai, Karapakkam, Chennai, Tamil Nadu, India

Keywords: Patient Data Security, Prescription, Encryption, AES Algorithm, Department-Specific Access, Secret Key, Secure Data Sharing, SHA-256 Encryption, Authorized Access, I Enter-Departmental Communication, Data Privacy.

Abstract: This paper introduces a safe solution for patient consultations for the healthcare industry. This includes recording medical information and encrypting the prescription with the Advanced Encryption Standard (AES) for safe communication from the Surgery, Radiology, and Pharmacy departments. Access is granted only to the concerned parties using decryption keys for every department, keeping the information safe. Outside of this, patient treatment updates in all the patients are stored in a secure fashion using SHA-256 hashing such that the information is safe against unauthorized modification. Performance metrics indicate the system is efficient with an average database storage retrieve time of 0.0002 seconds having low latency coupled with robust security. This two-layered security mechanism AES for encryption and decryption and SHA-256 for data storage enhances the protection for the confidentiality of patient data, the integrity of the data, and protection against unauthorized modification.

1 INTRODUCTION

The present healthcare system puts a premium on the secure sharing and storage of sensitive patient data. Digital transformation in healthcare introduces efficiency but risks exposing patient records, prescriptions, and medical notes to potential breaches. Robust encryption and hashing techniques are needed to ensure that sensitive information remains confidential, secure, and accessible.

In this framework, the Advanced Encryption Standard (AES), a widely adopted symmetric encryption algorithm, is employed to secure patient prescriptions. This is known for its speed, efficiency, and strong security attributes (J. Boddy et al. 2023). AES functions with fixed block sizes of 128 bits, supports key lengths up to 128 bits, 192 bits, and 256 bits. The attack upon this algorithm through brute force is not possible. This algorithm has been implemented in open-source libraries such as PyCrypto, OpenSSL, and Crypto++ (L.Suganthi et al.). In order to provide a higher degree of data integrity and ensure tamper-proof storage, SHA-256 has been used (Semantha et al. 2021). SHA-256 is part of the SHA-2 family and produces a fixed-length 256-bit hash, which depends completely on the input data. Any sort of change in data will produce entirely another hash, so any unauthorized change to the data will be easily identifiable. It is at the core of blockchain technology The cryptographic hash function is used to link data blocks in an immutable way (E.S. Selvapriya, et al. 2023). Platforms such as Open source Hyperledger and Ethereum use SHA-256 for blockchain-based solutions, which enable secure and transparent ways of managing data (Narasimha Rao, et al. 2024). In the proposed system, Patient will consult doctor and the doctor will upload the medical notes. AES encryption ensures the transfer of prescriptions between surgery, radiology, and pharmacy departments in a way that only the authorized staff, having the specific secret keys of their departments, can access the data (J. Boddy et al. 2023).

In parallel, SHA-256 is used for the encryption and storage of patient information in tamper-proof manner based on the block chain principle (M. Abaoud et al. 2023), thus allowing safe and unchangeable records as shown in Figure 1. Using open-source implementations and spring boot suite for backend and for data storage using MySQL, this

Priya, E. S. S., K., A., C., L. and S., A.
Analysis and Implementation of Security Algorithm for Healthcare Application.
DOI: 10.5220/0013891200004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 3*, pages 31-43
ISBN: 978-989-758-777-1

system saves on cost and takes advantage of the collective knowledge and continued improvement through the community (Aljoahni , et al. 2023). This paper introduces AES and SHA-256 as integrated to give an all-round secure framework in handling sensitive health data. The solution proposed will meet the needs of the health care industry while meeting the ever-growing demands for health data security (Mahadik et al. 2024).

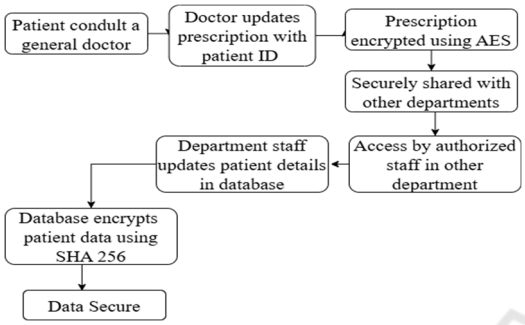

Figure 1: Block Diagram of System.

2 RELATED WORKS

There has been a lot of research done in integrating blockchain technology, encryption, and access control mechanisms into the management of healthcare data to overcome problems of data security, privacy, and interoperability in medical systems. It has been suggested that blockchain would be an effective means of securing medical records and ensuring data integrity in preventing unauthorized alterations and tampering, thereby making record management transparent (C. -L. Feng et al. 2023). Blockchain-based healthcare data sharing systems have been explored for the secure transmission of sensitive patient data across different healthcare providers while decentralizing and minimizing the risks of centralized storage (M. Abaoud et al. 2023). AES encryption has also been highly researched as a security technique for securing health information. This means that patient data will be secure and private between healthcare providers, as well as while storing sensitive medical records (M.A. Sahi , et al. 2017).

In a variety of applications, encryption will play a major role in securing medical data when sensitive patient information is being sent between parties. Role-based access control (RBAC) models are commonly used to ensure that only authorized personnel can access sensitive medical information. These models control access to patient data based on the roles and responsibilities of individuals within healthcare organizations. Implementing strict access control policies is critical to preventing unauthorized access to critical healthcare data (Jeyabose, et al. 2022). Other techniques considered are privacy-preserving techniques with blockchain, such as homomorphic encryption and zero-knowledge proofs, ensuring patient data will not be easily breached (Mahadik et al. 2024). All these techniques contribute to the realization of patient information security while safely transferring health records across various service providers. Finally, decentralized management through blockchain eliminates any single point failure, thus promoting safety by keeping away unauthorized people from manipulating health care records (Alzubaidy, et al. 2023). Blockchain-based approaches improve the Electronic Health Record interoperability so that information between the providers is exchangeable in an integrative and reliable way to ensure the integrity of medical records. This is set to be conducive to interoperability across the health entities as well as uniformity of the data between various systems (Aouedi, et al. 2022).

Blockchain has also been proposed to track the provenance of healthcare data, offering a transparent and auditable trail that enhances accountability in healthcare data management (Mahadik et al. 2024). Secure health information exchange via blockchain ensures the protection of patient privacy while guaranteeing data integrity and reducing the risk of data breaches. Smart contracts have high hopes toward the automation of the healthcare processes involved in data management. The system allows secure and efficient data sharing between healthcare providers while enforcing various privacy regulations with improvement in administrative tasks and showing stern security protocols (Tertulino et al. 2023). Recent research has illustrated the efficiency of Spring Boot and Java frameworks to develop secure, scalable, and high-performance health applications. The inherent security attributes of Spring Boot, support for RESTful API, and interoperability with MySQL have made Spring Boot a popular option for encrypting patient data (J. Wei, et al. 2022).

In addition, the Spring Security's RBAC pattern has also been widely researched to provide role-based access to medical information, maintaining HIPAA and GDPR compliance. Beyond this, Java-based applications provide asynchronous processing capabilities, which facilitate rapid encryption and decryption processes for real-time access to secured medical information without hitting system performance (Shraddha Dadhich, et al. 2016).

Research further shows that advanced indexing strategies in MySQL, for example, B-Tree and Hash Indexing, enhance the retrieval rate of encrypted patient information. Further, caching platforms like Redis and Earache have been incorporated in Spring Boot applications to improve performance with repeated database queries (Narasimha Rao, et al. 2024). Such studies indicate an increased interest and successful applications in using blockchain, encryption, and access controls in the management of healthcare data complexities. Therefore, such efforts toward making healthcare data decentralized yet transparent will be reflected in the proposed system (Nduma, et al. 2022).

3 PROBLEM STATEMENT

The healthcare sector has been aggressively digitalizing by the increased deployment of Electronic Health Records (EHR) as well as connecting various medical data management systems into one platform. Despite all of these technological advancements, critical challenges still surround the management and sharing of healthcare data, especially considering data security and patient privacy concerns as well as system interoperability (M.A. Sahi, et al. 2017).

The use of centralized systems in traditional health data systems poses high risks for breach, unauthorized access, and manipulation of private patient information. Such weaknesses can lead to large risks against both healthcare organizations and patients in the form of monetary losses, identity theft, and loss of reputation. Besides that, decentralization and isolation of health care systems pose the challenge of the secure exchange of patient data across different health providers and institutions (E.S. Selvapriya , et al. 2023).

It seeks to address all the above challenges using blockchain technology as the basis of a definitely safe, transparent, and interoperable system for the sharing, storing, and managing patient data. This is in relation to patient data security improvement using blockchain decentralized mechanisms and its cryptographical one. The immeasurability of data means that their integrity is always assured (Elmisery et al. 2016). The proposed idea combines blockchain, advanced encryption, and role-based access control models to achieve secure, patient-centered healthcare data management, with ultimate empowerment of the patient through rights over health records and unhindered, secured data exchange between health-care systems.

4 SYSTEM DESIGN

This paper provides the patient data protection system design for healthcare via AES-128 and SHA-256 encryption. The system protects patient prescriptions and medical records via secure storage, encryption, and access control by authorized staff in various hospital departments (J. Boddy et al. 2023). The system design has a front-end user interface written with HTML5, CSS, and JavaScript through which doctors and medical staff can access patients' records. The Java and Spring Boot-based back-end application performs the requests and handles the encryption process (S. Fugkeaw, et al. 2024). A MySQL database stores an encrypted patient record. Prescription data is AES-128 encrypted, and patient identifiers are SHA-256 hashed and secured against unauthorized tampering (Naresh, et al. 2023). Data flow starts from a patient visit general physician, and the physician will upload their prescription with an ID and medical notes. The prescription is then AES-128 encrypted and stored in the database. The prescription is then provided to authorized hospital staff from Surgery, Radiology, and Pharmacy departments depending on their role (M. Abaoud et al. 2023). Patient information is updated when modified and further secured using SHA-256 encryption prior to storage shown in Figure 2. The role- based authentication ensures that only approved medical staff are able to see and modify records, thus preserving data integrity and confidentiality. This system improves the protection of health data by incorporating AES-128 and SHA-256 encryption to protect sensitive patient information from unauthorized reading and modification and enable crucial medical personnel access to required information in a timely fashion (S. Fugkeaw , et al. 2024).

Software Integration: The software system is essentially an integration of multiple components functioning in consonance with each other to deliver efficient functionality and robust security. From a front-end point of view, the client application is communicating with back-end services using RESTful APIs, which present effective data transfer between the two sides, that of the client-side and server-side components. The responsive design ensures access on various devices while offering a friendly user experience on any platform. The system design is on top of the Java Spring Boot framework which brings in the power of all AES encryption and role-based access control, thus blockchain synchronization happening behind key business logic (Tertulino et al. 2023). As a result of this, this Spring

Boot has always taken great care to work on scalable maintainable architecture concerning sensitive data (Mahadik et al. 2024).

Applying SHA-256 integrates the blockchain, creating cryptographic hashes that help in building a decentralized and immutable ledger in which every transaction is written (Semantha et al. 2021). Data integrity is achieved through this because each blockchain entry validates corresponding database records, thus stopping tampering and modification of the record without authorization (C. -L. Feng et al. 2023). The layer that will be involved with the database will include participating in the storage of AES-encrypted records and blockchain hashes that accompany every entry to ensure data integrity remains sound.

These two technologies integrated into the system imply that data in the database are always synchronized with blockchain to reinforce security and authenticity within the system (Alzubaidy, et al. 2023). Two of them in combination help achieve a secure, efficient, and reliable solution in the management of sensitive information (M. Abaoud et al. 2023).

Existing Method: The modern healthcare system uses cloud computing for processing and storage of patient information. Though it facilitates remote diagnosis and easy access, it involves serious privacy concerns. Cloud servers are not absolutely secure, exposing patient information to hacking or misappropriation. Patients also have no direct authority over their health records, with cloud providers controlling access. The system relies mainly on k-Nearest Neighbor (kNN) and Mahalanobis Distance (MD) for pre-diagnosis, providing linear complexity without robust security mechanisms (Jeyabose , et al. 2022). While cloud systems are less energy-intensive than blockchain frameworks, they have the drawback of inefficient data exchange and distrusted, as patients and healthcare organizations are unable to completely trust cloud service providers with confidential medical data (S. Fugkeaw , et al. 2024).

Proposed Method: To solve this issue of privacy and security, the proposed system merges edge computing with blockchain technology. Instead of trusting cloud servers, patient data is processed locally at edge devices such as hospital PCs or personal devices, which ensures greater privacy (Tertulino et al. 2023). Blockchain technology provides an open and decentralized ledger for secure management of data access. Thus, Patients have complete control over their encrypted medical records and can grant access to only the authorized departments of healthcare (C. -L. Feng et al. 2023). The system employs AES encryption and SHA-256 hashing for providing enhanced security. Although blockchain is more energy-intensive, it significantly improves data trustworthiness, transparency, and resistance to illegal modifications (M. Abaoud et al. 2023). The approach eliminates the dependency on cloud storage, providing a more secure and dependable system of healthcare data management.

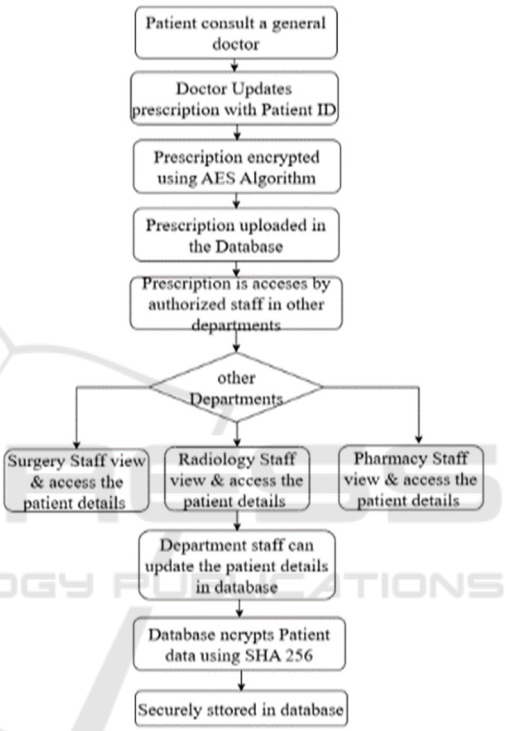

Figure 2: System Architecture.

5 METHODOLOGY

Data protection is a significant factor in contemporary healthcare systems to maintain the privacy, integrity, and authenticity of medical records. Sophisticated cryptographic methods like AES-128 (Advanced Encryption Standard - 128 bit) and SHA-256 (Secure Hash Algorithm - 256 bit) are used extensively to safeguard confidential medical data against unauthorized viewing and cyber attacks (Elmisery et al. 2016). AES-128 is a symmetric-key encryption algorithm for ensuring data confidentiality, whereas SHA-256 is a cryptographic hash function employed in ensuring data integrity. These crypto mechanisms complement each other to bolster the security of

healthcare applications through the encryption of patient data and the creation of immutable hash values for verification purposes (Aouedi, et al. 2022).

Advanced Encryption Standard (AES) Algorithm: The Advanced Encryption Standard (AES) algorithm encrypts data in blocks of 128 bits with the keys of lengths 128, 192, or 256 bits. AES starts with an Initial Round during which the plaintext is combined with the encryption key using the step Add RoundKey (J. Boddy et al. 2023). This is followed by a number of Main Rounds, each with four phases: SubBytes (byte substitution using an S-box), ShiftRows (cyclical shift of the rows for diffusion), MixColumns (mixing columns with matrix multiplication for further diffusion), and AddRoundKey (mixing of the state with a round-key). The number of rounds depends on the key size: 10 rounds for keys with size 128 bits, 12 rounds for keys with size 192 bits, and 14 rounds for keys of size 256 bits as shown in Figure 3. In the Final Round, the MixColumns operation is omitted, thereby leaving the ciphertext still 128-bit block-sized (Nduma , et al. 2022). The layered implementation of AES provides outstanding security against cryptographic attacks without sacrificing high performance (M.A. Sahi , et al. 2017).

Secure Hash Algorithm-256(SHA-256): SHA-256, ahash, and phash are hash algorithms employed for many applications. SHA-256 is a cryptographic hash function that minimizes input data to a fixed 256-bit hash value, providing data integrity and security through intricate mathematical processes such as bitwise shifts, rotations, and modular additions (Semantha et al. 2021). SHA-256 finds extensive application in digital signatures, certificates, and block chain because it is resistant to collision and pre-image attacks ((L.Suganthi et al.). Conversely, ahash (Average Hash) and phash (Perceptual Hash) are intended for image matching. Ahash scales down an image, converts it to grayscale, and creates a binary hash based on average pixel values, and hence is sensitive to small variations (M. Abaoud et al. 2023). Phash uses the Discrete Cosine Transform (DCT) to detect low-frequency components that define the overall shape of the image, offering greater resistance to rotation and scaling as shown in Figure 4. Although SHA-256 provides cryptographic security, ahash and phash are good at identifying perceptual similarities and are hence applicable in digital forensics as well as image deduplication (J. Boddy et al. 2023).

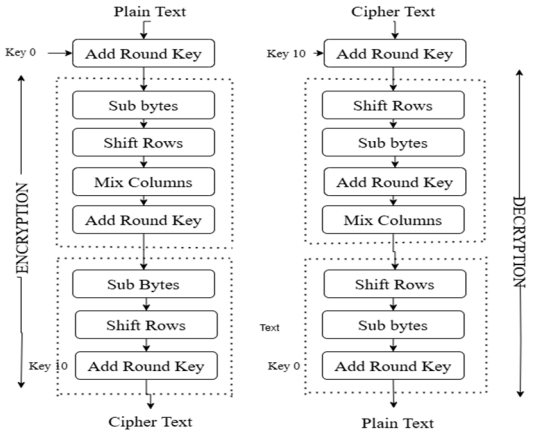

Figure 3: AES Flowchart for Encrytion and Decryption in 128 Bites.

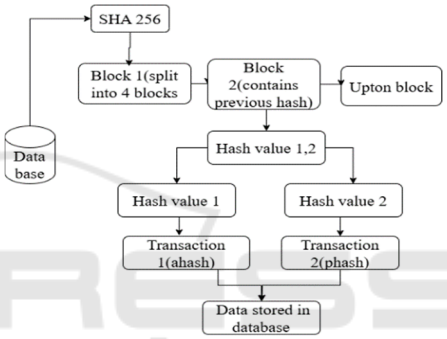

Figure 4: SHA 256 for Encryption.

6 RESULT AND DISCUSSIONS

The website setup is in place to validate both the security and functionality and performance of a six-module website-based system, namely Admin, User, Doctor, Surgeon, Radiologist, and Pharmacist as shown in Figure 7. The Admin module can access directly with login functionality as shown in Figure 8. All other modules require approval from the Admin before the use of login entry to the system as shown in Figure 9. This strictly integrates role-based access control. The data is encrypted by AES-128 with symmetric keys, creating the ciphertext for preserving the confidentiality of the data stored and transmitted.

In addition to this, the blockchain based on SHA-256 will be added in order to create a decentralized and immutable layer of security and protect the integrity of the data by preventing tampering. MySQL has been used for the database storing encrypted records by AES and their hashes in the blockchain

The encryption workflow was validated with tests on generation and handling of symmetric keys and

ciphertext. The blockchain nodes were deployed on distributed servers for redundancy, during which synchronization tests were executed under simulated partition of the network and concurrent transactions. Scheduled audits would cross-check the blockchain hashes with MySQL database records and presented tampering scenarios that validated the system's capability to detect any unauthorized modifications.

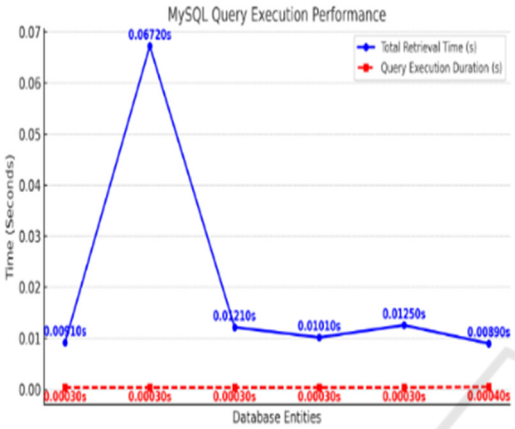

Figure 5: MySQL query execution time for fetched data across all departments, including total time taken and entities (departments).

Table 1 illustrates the comparison between retrieval time and query execution time for various database entities. The six database entities specified in the table are Doctor Entity, Patient Entity, Pharmacy Entity, Radiologist Entity, S-Doc Request Table, and User Table. Number of Records Retrieved refers to the number of records returned for each of the tables in the retrieval process. Total Retrieval Time (seconds) is the time it takes for the respective records to be retrieved in the data-base, while the Query Execution Duration (seconds) is the time it takes for the respective query to be processed by the data-base as shown in Figure 5.We can observe the highest retrieval time of 0.0672 seconds taken by the Patient Entity due to the relatively higher number of records (15) in comparison to the other entities. The Doctor Entity and User Table take retrieval times of 0.0091 seconds and 0.0089 seconds respectively due to the fewer records. The retrieval time of the S-Doc Request Table is merely 0.0125 seconds despite the higher number of records (29). This is good performance of the database for large data sets. Query execution time for each of the tables is the same at approximately 0.0003 seconds with the User Table timing a bit higher at 0.0004 seconds.

Table 1: Query execution performance for healthcare of each database entities.

Database Table	Number of Records Retrieved	Total Retrieval Time (s)	Query Execution Duration (s)
Doctor Entity	8	0.0091	0.0003
Patient Entity	15	0.0672	0.0003
Pharmacy Entity	2	0.0121	0.0003
Radiologist Entity	2	0.0101	0.0003
S-Doc Request Table	29	0.0125	0.0003
User Table	12	0.0089	0.0004

The difference can be due to the indexing or query optimization internally for different tables. Table 1 performance result illustrates the performance of the database in retrieving varying amounts of data for various entities with minimal query execution time. The efficient data retrieval ensures quick data accessibility for the respective healthcare management modules, thereby ensuring effective performance operations with minimal response time.

Figure 6: Local server setup.

Figure 6 shows the local server setup using Spring Boot. The server.port=8099 configures the application to run on port 8099, accessible via http://localhost:8099. The database connection is established using a MySQL database named at on port 3306.

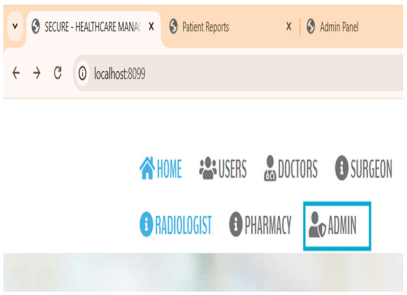

Figure 7: Functional modules of the system.

The system has six major modules, and they are as shown in Figure 7

- User: This module is for users or patients. User when comes to the hospital they will Register & Login in the hospital website.
- Doctors: This module is for medical specialists or general physicians. Doctors provide a consultation to users and they will upload the patient details and it will be saved in the database.
- Surgeon: Surgeons can use this module with the features such as scheduling surgery, viewing pre and post-surgery reports, interacting with radiologists and doctors, and changing patient recovery status.
- Radiologist: Radiologist Module is designed for the radiology specialists to deal with medical images. It allows them to upload and arrange reports, interpret the outcome of the scan, and give the result to doctors and surgeons. It helps in more proper diagnoses and treatments.
- Pharmacy: This module facilitates the management of medicines and prescriptions, such as inventory control, prescription processing, medicine expiry dates tracking, and prescription coordination with doctors.
- Admin: Admin has complete authority over the system and can directly log into any module. Handling Users, Doctors, Surgeons and Radiologists. Admin will give the permissions to the other departments for further treatment to the user.

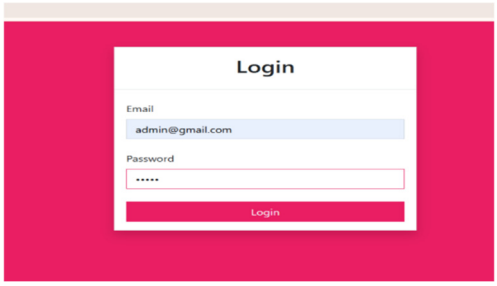

Figure 8: Admin user interface.

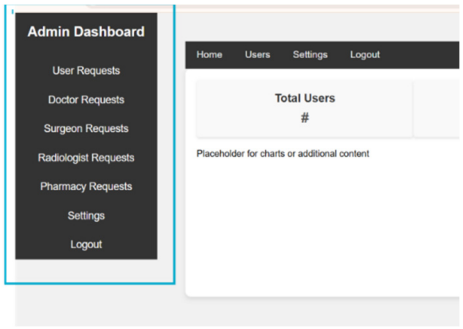

Figure 9: Admin dashboard with modules.

Admin login: The admin should enter their registered email (e.g., admin@gmail.com and respective password as shown in Figure 8. Admin will be having direct hospital portal and give access to the other departments and this allows admin doesn't need separate login for each department and can take the control of all department functionality into single account.

The Admin Dashboard is the main interface from which the administrator can manage and monitor all the system activities as shown in Figure 9. The dashboard grants access to various modules and enables the admin to process user requests effectively.

- User Request: Display and manage requests from general users.
- Doctor Request: Process requests from doctors.
- Surgeon Requests: Process surgeon-related requests.
- Radiologist Request: Monitor requests from radiologists.
- Pharmacy Request: Monitor pharmacy-related requests.
- Settings: Set up system settings.
- Logout: Log out of the admin portal securely

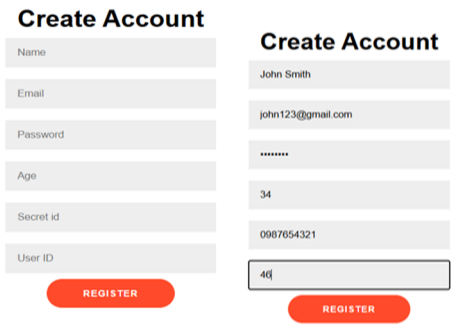

Figure 10: Step-1: User account create page and step 2: user registration.

The User Registration page will allow new users to create an account in the hospital website as shown in Figure 10. This process will ensure secure access and authentication for all modules. The steps for registration are shown in Figure 10.

Steps in the User Registration Process:

- Name: Full name of the user.
- Email: A valid email address for account verification.
- Password: A secure password for login authentication.
- Age: The user's age, used for profile identification.
- Secret ID: Users contact number for any verification.
- User ID: A system-generated or manually entered unique identifier for the user.

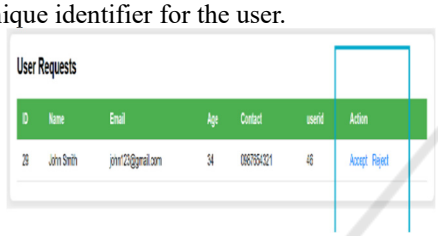

Figure 11: User request to admin.

The request is received by the admin under the User Requests section. The admin has two choices: Accept the request (provide Access to the ssystem). Deny the request (deny access). The decision is reflected in the system, updating the user's status as shown in Figure 11. When the admin checks the request, the information is saved in the database. If granted, the user information is stored for good. If refused, the request may be logged but denied authorization. The database is used to keep a record of all the requests made by the users to provide security and convenience. If the request is granted, the user is issued credentials to access as shown in Figure 12. If denied, the user has to reapply with correct information. The admin validates that only rightful users receive access. This procedure allows for safe management of users within the hospital portal system.

Figure 12: User login.

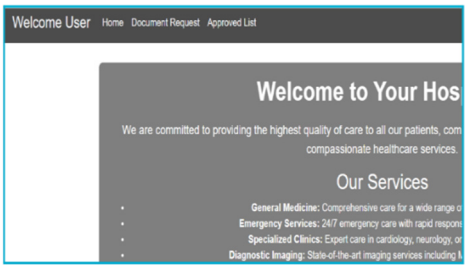

Figure 13: User dashboard with document request and user authentication.

The system verifies the data that is stored in database. if the data are matching the user will successfully be logged in and redirect to their dashboard.

- User Dashboard contains three part as shown in Figure 13.
- Home Section: Displays general information about related to hospital services.
- Document Request: Allows users to request their specific document/files related to medical history and submit the request to the related departments those who uploaded.
- Approval List: Displays the status of pending or approved document requests by the respective department. The user can view the status of their requests. The admin reviews the request and either approves or rejects it. Approved documents become available for download or further processing.

Figure 14: Step 1-Doctor register page and Step 2-Doctor sign up and Step 3- Doctor login after admin approval.

The doctor will register and submit request to the admin, providing their details such as name, email, password, specialty, proof and contact information as shown in Figure14 (a) and Figure (b). The admin reviews the request and either approves or rejects it. If approved, the doctor can sign in using their registered email and password as shown in Figure 14(c). Once logged in, the doctor can view patient

requests for medical consultations, prescriptions, or reports. The doctor can choose to accept or reject a request. If accepted, the details are stored in the database, and the user gets access to the approved documents or services. This ensured robust encryption and proper treatment of data while ensuring operational dependability, thus allowing for a user-friendly yet secure system for all the modules.

The system meets the requirements for confidentiality, integrity, and security of data, with the incorporation of AES-128 encryption, SHA-256 block chain, MySQL for data storage, and access control mechanisms, and yet remains scalable and high in performance. The system was tested on performance, security, and functionality; results indicate that it can indeed handle real-world operational environments in managing sensitive healthcare data.

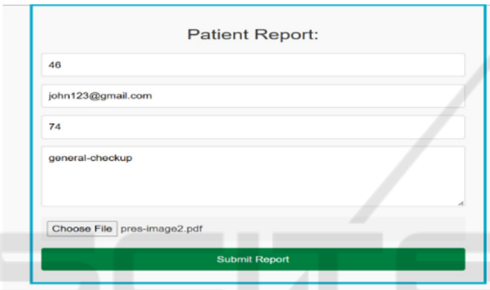

Figure 15: Doctors will upload the patient details.

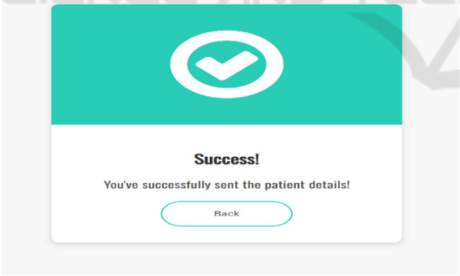

Figure 16: Patient report submission confirmation.

After approving a user's request, the doctor fills in the patient details and report description in the system. The doctor can also attach necessary medical files (e.g., prescriptions, lab results, or scanned documents) as shown in Figure 15 and sample file as shown in Figure 19. Once all required information is entered, the doctor submits the report by clicking the "Submit Report" button. After successful submission, the system processes the report and stores it in the database.

A confirmation message is displayed, indicating that the patient details have been successfully sent. The user (patient) can now access the uploaded report through their dashboard as shown in Figure 16.

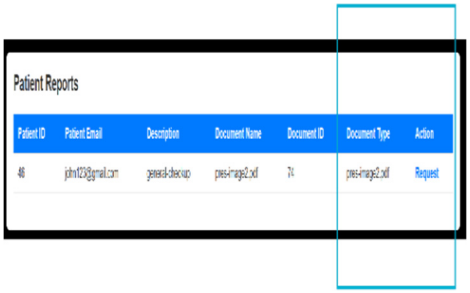

Figure 17: Patient report in their dashboard asking for a request to see the document.

After prescription uploaded by the respective doctor, the user should request to view and download the document, which will be displayed in the user dashboard as shown in Figure 17.

Figure 18: After request enter the specific key value.

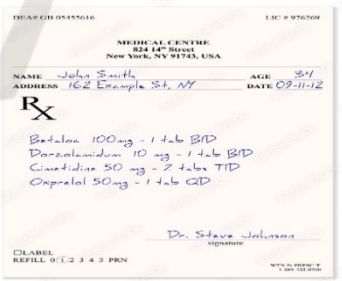

Figure 19: Sample prescription (input).

After the request has been approved, the system prompts the patient to enter his or her own distinct ID and key value. These are employed as an authentication factor to make certain that only the respective patient gains access to his or her report. Upon correct entry of ID and ke value, the patient can proceed to safely view or download his or her medical report as shown in Figure 18. The uploaded document

serves as input data that will later be used to verify as shown in Figure 19 and check decryption results.

The Advanced Encryption Standard (AES-128) is utilized to encrypt sensitive data before it is stored in the database. Encrypting confidential patient information in this way makes it accessible only to authorized individuals with the correct decryption key. This aspect made it possible for information to be transmitted securely between various modules, including Admin, Doctor, Surgeon, Radiologist, and Pharmacist, without compromising the responsiveness of the system as a whole.AES-128 encryption provided the level of security needed without burdening the system with computational overhead and therefore was the ideal solution to encrypt health information.SHA-256 (Secure Hash Algorithm) is used to generate a fixed-size hash of information. It aids in data integrity verification to ascertain that data stored is not compromised or altered. Unlike AES, SHA-256 is a one-way function that implies the hashed data cannot be decrypted back to its original form. The encrypted messages (AES-128) and hashed values (SHA-256) are stored in the database, which is a secure way of protecting patient records against unauthorized access.

Blockchain integration via SHA-256 hashing algorithm improved the security with greater transparency and openness. It would generate, with each change, a cryptocurrency hash that the node would attach to the blockchain. The system utilized blockchain to guarantee data immutability for ensuring resistance against data tampering and alterations. As demonstrated in testing with load scenarios, blockchain can sync seamlessly at distributed nodes, without introducing additional delay in validations and recording the transaction. This featured decentralized storage and ensured availability of data while preventing single points of failure. Any attempt to modify or manipulate blockchain records would be immediately detectable, thus securing the integrity of healthcare information and providing transparency into all actions performed within the system. MySQL was used as the database management system, therefore storing encrypted healthcare records secure as shown in Figure 20. This ensured a very smooth interface of encrypted data with blockchain hashes, thus making it efficiently handle huge amounts of data. The database was kept synchronized with the blockchain records, which were bound to the entries in MySQL. Therefore, the system guaranteed easy verification of authenticity of health records. Performance of MySQL in terms of a high load was good. Data retrieval or storage did not introduce noticeable delays in the operations. Scalability would allow this system to be scaled to very large volumes of health care data when users and health care records became many in numbers.

Encryption Process: The input data as shown in Figure 19 (likely an image, as mentioned) has been encrypted using AES-128 (Advanced Encryption Standard - 128-bit). AES-128 is a symmetric encryption algorithm that ensures data confidentiality by converting readable data into an unreadable format. The encrypted message appears as a randomized string of characters, making it incomprehensible without the correct decryption key.

Access control mechanisms were established and only permitted approval users to access the system. There existed a procedure for admin approval prior to the issue of access to a user on the platform; thus, limited entry by access granted to Doctors, Surgeons, Radiologists, and Pharmacists, therefore not allowing unauthorized users to access sensitive information of health care by preventing them from getting into the system.

Figure 20: Encrypted message of input (image) in AES 12.

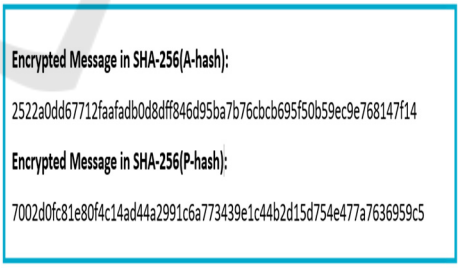

Figure 21: Encrypted Message in Sha 256 in A-Hash and P-Hash.

SHA-256 (Secure Hash Algorithm - 256 bit) is a one-way cryptographic hashing function encrypted message as shown in Figure 21, meaning it cannot be decrypted back into its original form. Access logs were also maintained and blockchain recorded the same, hence transparency and accountability of user activities in the system were ensured. The system

proved strong scalability regarding performance under load.

Figure 22: The approved list section shows the document requests that have been approved by the doctor.

The Approved List Section in the healthcare system with approved document requests from the doctor as shown in Figure 22. The table has important data such as Patient ID, Doctor ID, Department, Description, File Key, and Action. After the approval of a request, a File Key (decryption key) is created, enabling authorized users to view the document. Action in the column consists of a Download option, as shown in Figure 23 and users can acquire the approved document securely. In this process, only authenticated and authorized users access patient-related details, thereby authorized users access patient-related details, thereby augmenting data confidentiality.

Figure 23: Once approved, the File Key (decryption key) is generated randomly.

The automatic creation of a File Key (decryption key) after the document request has been authorized. The key is created randomly and serves as a security token, with only authorized users being able to decrypt and view the document as shown in Figure 23. This feature improves data privacy by limiting unauthorized users and protecting sensitive medical records. Both key-based access and document authorization provide a safe and managed document-sharing environment within the system.

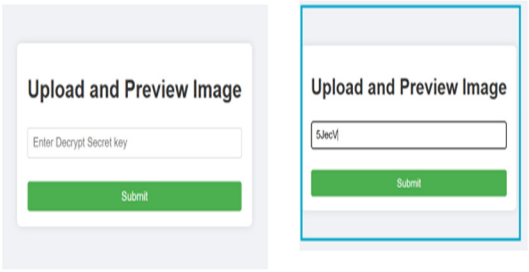

Figure 24: After clicking download a pop-up box will and after entering the decrypt key (file Key) and submit it will show the decrypted file open for View the file and for decrypt key.

The File Decryption Interface, which is in the form of a pop-up box when the user selects the download option. The interface asks for a Decrypt Secret Key from the user in order to open the file. The key is generated beforehand at the document approval process and is kept safe in the database. When the decryption key is entered, the system authenticates it by comparing it with the key stored in the database as shown Figure 24 (a) and Figure 24 (b). If the key is correct, the system decrypts the file using the AES-128 decryption algorithm and displays it. This approach provides a secure mechanism where the confidential information can be made available only to authorized users with the correct decryption key, thus improving data security and privacy. In the event of an invalid key, the system prompts for an error message to avoid unauthorized access. The correct decryption key needs to be re-entered by the user in order to access the file. This strategy enforces data security using AES-128 encryption to maintain safe and controlled access to the sensitive data the decrypted medical prescription upon successful decryption with the AES-128 algorithm as shown in Figure 25. When the decryption key is entered correctly, the system displays the patient's details, medical reports. This protects sensitive medical data from unauthorized access while ensuring only legitimate users access the information at the same time

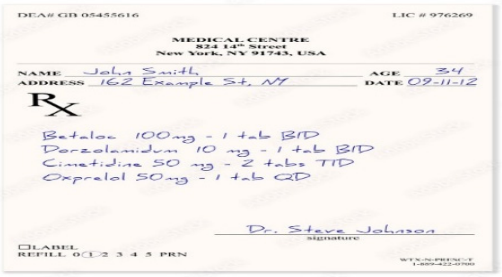

Figure 25: Decryption file (output).

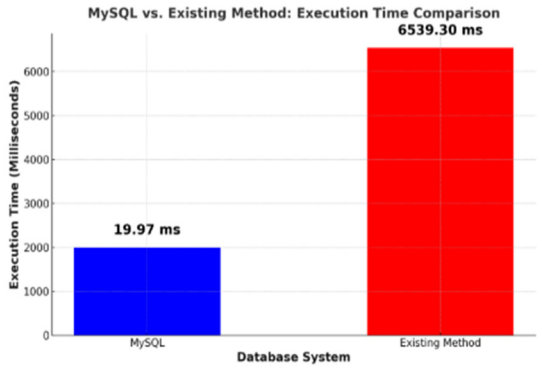

Figure 26: Comparison of MySQL execution time Vs Existing method execution time (Avg execution time)

Table 2: MySQL vs. Existing Method Query Execution Time Comparison.

System	Average Execution Time (Seconds)	Performance Observation
MySQL (Avg Execution Time)	0.01997s	Extremely Fast
Existing Method (Avg Execution Time)	6.5393s	Significantly Slower

The table 2 shows that MySQL (0.01997s) is faster than the existing method (6.5393s), making it ideal for real-time applications. The existing method's slower execution suggests inefficient queries or indexing, requiring optimization to improve performance. Overall, MySQL is the better choice for speed and efficiency A comparision graph for this is as shown in Figure 26. It easily supported hundreds of concurrent users with a minimal decrease in response time or system performance. The database, encryption operations, and blockchain synchronization all scaled effectively without much loss of performance. It was possible to process hundreds of thousands of encrypted healthcare records at a very low response time. The decentralized nature of the blockchain network did not cause bottlenecks; transactions were validated properly even under heavy loads. This ensures the system can achieve large healthcare networks where multiple users are accessing and updating records in real time.

7 CONCLUSIONS

The current research was capable of developing a secure healthcare management system based on AES-128 encryption, SHA-256 blockchain technology, and MySQL for storage. The design of the system offers robust security for sensitive health information through stressing confidentiality, integrity, and transparency. AES-128 encryption covers the data when stored and transferred, while blockchain offers an irreversible record of transactions, thus rendering the system's data integrity secure. MySQL was used to store the encrypted data securely with consistency, as well as facilitating easier verification. Comparing performance between the MySQL database and the existing method demonstrated a vast difference in execution time. The MySQL database executed queries in just 19.97 ms execution time, which was significantly lower compared to the 6539.30 ms by the existing method. This performance gap clearly demonstrates the higher efficiency of MySQL in handling and processing data queries. The steep drop in execution time reflects MySQL's ability to provide faster access to data, which is totally required by real-time healthcare applications.

The outcome is that the proposed system not only satisfies healthcare data security requirements but also optimizes system performance by a considerable extent. With MySQL's improved execution times, the solution is now ideal for real-time applications and offers quick, secure access to healthcare records. In conclusion, the use of AES-128 encryption, SHA-256 blockchain, and MySQL is an efficient and scalable method for secure and effective healthcare data management.

REFERENCES

A. J. Boddy, W. Hurst, M. Mackay and A. e. Rhalibi, "Density-Based Outlier Detection for Safeguarding Electronic Patient Record Systems," in IEEE Access, vol. 7, pp. 40285-40294, 2019, doi: 10.1109/ACCESS.2019.2906503.

Aljoahni, Tahani & Zhang, Ning. (2023). Secure, ID Privacy and Inference Threat Prevention Mechanisms for Distributed Systems. IEEE Access. PP. 1-1. 10.1109/ACCESS.2023.3234932.

Alzubaidy, Hussein & Al-Shammary, Dhiah & Abed, Mohammed & Ibaida, A. & Ahmed, Khandakar. (2023). Hilbert Convex Similarity for Highly Secure Random Distribution of Patient Privacy Steganography. IEEE Access. PP. 1-1. 10.1109/ACCESS.2023.3325754.

Aouedi, Ons & Sacco, Alessio & Piamrat, Kandaraj & Marchetto, G. (2022). Handling Privacy-Sensitive Medical Data with Federated Learning: Challenges and Future Directions. IEEE Journal of Biomedical and Health Informatics. PP. 1- 14. 10.1109/JBHI.2022. 3185673.

C. -L. Feng, Z. -C. Cheng and L. -J. Huang, "An Investigation into Patient Privacy Disclosure in Online

E.S. Selvapriya and L. Suganthi. 2023. Design and implementation of low power Advanced Encryption Standard cryptocore utilizing dynamic pipelined asynchronous model. Integr. VLSI J. 93, C (Nov 2023). https://doi.org/10.1016/j.vlsi.2023.102057

Elmisery, Ahmed & Rho, Seungmin & Botvich, Dmitri. (2016). A Fog Based Middleware for Automated Compliance with OECD Privacy Principles in Internet of Healthcare Things. IEEE Access. 4. 8418-8441. 10.1109/ACCESS.2016.2631546.

J. Wei, X. Chen, J. Wang, X. Hu and J. Ma, "Enabling (End-to-End) Encrypted Cloud Emails with Practical Forward Secrecy," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 4, pp. 2318- 2332, 1 July- Aug. 2022, doi: 10.1109/TDSC.2 021.3055495.

Jeyabose, Andrew & Karthikeyan, J. & Andrew, Jennifer & Pomplun, Marc & Dang, Hien. (2022). Privacy Preserving Attribute-Focused Anonymization Scheme for Healthcare Data Publishing. IEEE Access. PP. 1-1. 10.1109/ACCESS.2022.3199433.

L.Suganthi, R.Anandha Praba , E.S.Selva Priya. “Enhanced Arrhythmia Detection Through Wavelet Scattering and Deep Learning Techniques”, Journal of University of Shanghai for Science and Technology, ISSN: 1007-673

M. Abaoud, M. A. Almuqrin and M. F. Khan, "Advancing Federated Learning Through Novel Mechanism for Privacy Preservation in Healthcare Applications," in IEEE Access, vol. 11, pp. 83562-83579, 2023, doi: 10.1109/ACCESS.2023.3301162.

M.A. Sahi et al., "Privacy Preservation in e-Healthcare Environments: State of the Art and Future Directions," in IEEE Access, vol. 6, pp. 464-478, 2018, doi: 10.1109/ACCESS.2017.2767561.

Mahadik, Shalaka & Pawar, Pranav & Muthalagu, Raja & Prasad, Neeli & Hawkins, Sin-Kuen & Stripelis, Dimitris & Rao, Sreedhar & Ejim, Peter & Hecht, Bruce. (2024). Digital Privacy in Healthcare: State-of-the Art and Future Vision. IEEE Access. PP. 1-1. 10.1109/ACCESS.2024.3410035.

Medical Platforms," in IEEE Access, vol. 7, pp. 29085-29095, 2019, doi: 10.1109/ACCESS.2019.2899343.

Narasimha Rao, K. P., & Chinnaiyan, S. (2024). Blockchain-Powered Patient-Centric Access Control with MIDC AES-256 Encryption for Enhanced Healthcare Data Security. Acta Informatica Pragensia, 13(3), 374-394.

Naresh Sammeta,Latha Parthiban, "Blockchain-based Scalable and Secure EHR Data Sharing using Proxy Re-Encryption", The International Arab Journal of Information Technology (IAJIT) ,Volume 20, Number 05, pp. 702 - 710, September 2023, doi: 10.34028/iajit/20/5/2.

Nduma, Basil & Ambe, Solomon & Ekhator, Chukwuyem & Fonkem, Ekokobe. (2022). Health Records Database and Inherent Security Concerns: A Review of the Literature. Cureus. 14. 10.7759/cureus.30168.

S. Srilaya and S. Velampalli, "Performance Evaluation for DES and AES Algorithms- A Comprehensive Overview," 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India,2018, pp. 1264- 1270, doi: 10.1109/RTEICT42 901.2018.9012536.

S. Fugkeaw, L. Hak and T. Theeramunkong, "Achieving Secure, Verifiable, and Efficient Boolean Keyword Searchable Encryption for Cloud Data Warehouse," in IEEE Access, vol. 12, pp. 49848-49864, 2024, doi: 10.1109/ACCESS.2024.3383320.

Semantha, Farida Habib & Azam, Sami & Shanmugam, Bharanidharan & Yeo, Kheng Cher & Beeravolu, Abhijith. (2021). A Conceptual Framework to Ensure Privacy in Patient Record Management System. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3134873.

Shraddha Dadhich "Performance Analysis of AES and DES Cryptographic Algorithms on Windows & Ubuntu using Java". International Journal of Computer Trends and Technology (IJCTT) V35(4):179-183, May 2016. ISSN:2231-2803. www.ijcttjournal.org. Published by Seventh Sense Research Group.

Tertulino, Rodrigo & Ivaki, Naghmeh & Morais, Higor. (2024). Design a Software Reference Architecture to Enhance Privacy and Secu in Electronic Health Records. IEEE Access. 12. 112157. 10.1109/ACCE SS.2024.3441751.

Wahyudi, R., & Romli, M. A. Android-based Patient Medical Record Data Security Application using AES and RSA Method Cryptography. International Journal of Computer Applications, 975, 8887.