

PRIVACY PRESERVING DATA SHARING IN CLOUD-BASED HEALTHCARES SYSTEMS

1st G. Laxmi prasanna, 2nd A.Haritha, 3rd D. Harini 4th Dr. G. Victo Sudha George, 5th Dr.M.Chandran ,

6th Mr.C.Balaji

1,2,3UG Student, Department of Computer Science and Engineering, Dr. MGR Educational and Research Institute, Maduravoyal, Chennai 600095, TN, India

4th Additional HOD, Department of Computer Science and Engineering Dr. MGR Educational and Research Institute, Maduravoyal, Chennai 600095, TN, India

5,6Professor, Department of Computer Science and Engineering Dr. MGR Educational and Research Institute, Maduravoyal, Chennai 600095, TN, India

mahendergundapu123@gmail.com, avulaharitha19@gmail.com, hariniduraimurugan2004@gmail.com

Abstract

Securing confidential data in cloud-based health systems is solely critical for the safe storage and exchange of personal patient data. Techniques like encryption, access control methods, and data anonymization limit unauthorized access and facilitate the open flow of information. Blockchain and secure protocols for communication serve to establish confidence, regulatory compliance, and data integrity in health systems. A hybrid framework utilizing blockchain and high-end encryption methods is postulated in this paper for optimizing data security along with system efficiency. The designed model is optimized through some performance metrics which reflect its capability to effectively weigh security against the system performance. Data Integration and Collection to allow easy merging of healthcare data from various sources, Data Encryption for the protection of sensitive data in transit and at rest, Access Control and Authentication for proper control of user access and confidentiality, Data Anonymization and Pseudonymization for patient identity protection, and Secure Protocols for Safe Data Sharing to provide secure and controlled data sharing among stakeholders. The framework seeks to achieve a balance between data protection and data accessibility in order to instill trust in cloud-based healthcare systems and ensure the ethical use of medical data for better healthcare outcomes.

Keywords— Privacy-preserving, Data encryption, Access control, Data anonymization, Secure data sharing, Blockchain in healthcare.

I. Introduction

With the advent of the digital age, cloud-based healthcare platforms have changed the manner in which patient data is handled, stored, and accessed. Although these systems are rich in benefits like scalability, cost-effectiveness, and enhanced data availability, they are also vulnerable to numerous cybersecurity threats. Personal health records (PHRs), electronic health records (EHRs), and other confidential medical data contained within these platforms need to be strongly protected against unauthorized access, data breaches, and Challenges to Health System Security: Centralized Vulnerabilities: Healthcare systems traditionally hold sensitive information on centralized servers, which are sitting ducks for hackers .Privacy Risks: Improper access to patient information can result in identity theft, fraud, and abuse of personal data. Compliance Requirements: Medical facilities are required to adhere to strict regulations like the Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR) to preserve the privacy and security of the patient data.

II. Literature Survey

[1] Strengthening Patient Data Privacy in Cloud Healthcare Systems This research explores multiple strategies of encryption like homomorphic encryption and attribute-based encryption to protect the patient data present in

the cloud. The article points out the difficulty of reconciling security and data usability, suggesting a hybrid model that maintains confidentiality of patients while allowing controlled access to data for research. [2] A Blockchain Framework for Secure Sharing of Health Data. The paper discusses blockchain integration in cloud healthcare for guaranteeing tamper-proof data sharing. It elaborates how decentralized management of data improves security, avoids unauthorized access, and builds trust among stakeholders. It also points out scalability issues while dealing with vast amounts of data. [3] Privacy-Preserving Access Control Mechanisms in Healthcare Clouds. This study aims to create multi-level access control systems through role-based and attribute-based access controls. The research prioritizes reducing data exposure by providing access rights based on users' roles and authentication levels, providing strong security and regulatory compliance in the healthcare sector. [4] Differential Privacy Techniques for Cloud-Hosted Medical Data. This research suggests the implementation of differential privacy algorithms to anonymize sensitive medical information prior to uploading it to the cloud. It examines privacy versus data accuracy trade-offs, demonstrating that introducing controlled noise into datasets preserves patient confidentiality while ensuring data utility for analysis. [5] Secure Data Sharing Protocols in Healthcare Clouds. The paper introduces a new secure data sharing protocol using public-key infrastructure (PKI) and digital signatures. The new method guarantees data integrity and authentication when transmitted, which avoids unauthorized use and ensures only trusted entities have access to patient records.

III. Domain

This workflow focuses on the secure management of healthcare data in the Cloud environment with respect to privacy of patients. Data is collected by patients & devices and shared in secure modes. The provided data is encrypted during transmission and storage. Anonymization method for protection of patients' identity is used, before protected data goes into Cloud Storage. Protected data gets accessed by authorized Data Consumers through authenticated procedures to ensure privacy and security. The field of privacy-preserving sharing of data in cloud-based healthcare is concerned with protecting the confidentiality and security of patient information while facilitating efficient access by authorized users. With the changeover of healthcare systems to digital mode, secure storage, transmission, and sharing of data have become imperative. Scalable, cost-effective solutions are offered by cloud computing, yet privacy issues are raised by the threat of unauthorized access, data breaches, and regulatory compliance issues. In this field, patients are the owners of the data, creating medical records from hospital visits, diagnostic tests, and wearable health devices. Hospitals and clinics keep these records in electronic health systems, frequently combining data from multiple sources, such as IoT medical devices that monitor health metrics continuously. For security purposes, the cloud storage system is used to store encrypted data, avoiding unauthorized access.

IV. EXISTING SYSTEM

In today's cloud-based healthcare systems, keeping sensitive patient data confidential and secure is a top priority. Most existing systems depend on traditional encryption methods, basic access controls, and data anonymization techniques to protect personal health information. Commonly used encryption algorithms like AES and RSA help secure data both while it's being transmitted and when it's stored. However, these encryption methods often require decryption for processing, which can leave data vulnerable to breaches. To manage access to sensitive information, Role-Based Access Control (RBAC) and Discretionary Access Control (DAC) models are frequently used. RBAC limits access based on specific roles within the organization, while DAC gives data owners the power to grant or revoke access permissions. Unfortunately, these models can lack the flexibility needed in dynamic environments and may struggle to keep up in large healthcare systems. Data Anonymization Techniques, such as k-anonymity and data masking, are employed to protect patient privacy while still allowing for data sharing in research and analysis. While these methods help reduce the chances of re-identification, they can sometimes result in information loss or inadequate protection against inference attacks. To ensure secure communication between healthcare entities, Transport Layer Security (TLS) protocols are used to encrypt data as it travels over networks. However, relying solely on TLS doesn't safeguard against internal threats or data breaches that can occur due to compromised access credentials. Additionally, Centralized Storage Systems are mainly used to house patient data, which creates a single point of failure and vulnerability. If unauthorized access or data breaches happen in these centralized setups, it can lead to significant data exposure. Plus, keeping track of audit trails and meeting regulatory standards like HIPAA and GDPR continues to be a challenge for existing systems. Even with these security measures in place, many current systems still find it tough

to strike the right balance between making data accessible, maintaining performance, and staying compliant with regulations. This situation really underscores the necessity for more sophisticated solutions like blockchain, homomorphic encryption, and decentralized access control to effectively manage healthcare data while keeping.

V. RESEARCH GAP

Even though encryption, access control mechanisms, and data anonymization techniques are becoming more common to protect patient information in healthcare systems, there are still some significant gaps that need to be addressed: Blockchain technology has the potential to greatly enhance security and data integrity, but its actual use in healthcare systems is still just getting started. We need more research on scalable blockchain architectures that can manage large amounts of healthcare data while keeping latency low and efficiency high. While encryption and access controls do help secure data, the challenge lies in enabling real-time secure data sharing among healthcare providers without putting patient privacy at risk. Current systems often lack effective protocols that can balance data accessibility with the need to protect privacy. Many existing RBAC models fail to fully consider the dynamic and hierarchical nature of healthcare workflows, which can lead to either too much access or overly restrictive permissions. There's a pressing need for research to create adaptive access control models that can adjust to changing roles and contexts in healthcare environments. While data anonymization techniques help lower the risk of re-identification, they often struggle to maintain data utility while providing strong privacy protections. We need to develop more advanced algorithms that can effectively balance privacy with the usability of data.

VI. PROPOSED SYSTEM

1. System Overview

The system we're proposing is a cutting-edge healthcare data management platform designed to be both secure and scalable. It uses the latest web technologies and strong security protocols to keep sensitive healthcare information safe, ensuring that confidentiality, integrity, and availability are always prioritized. Our goal is to make it easy to store, retrieve, and share healthcare reports and documents, all while implementing role-based access control to meet data security standards

2. Objectives

- Create a user-friendly and responsive interface for both healthcare professionals and patients.
- Implement secure user authentication and control over data access.
- Protect data with end-to-end encryption and secure storage solutions.
- Facilitate efficient document sharing with real-time access control and audit logging.
- Ensure smooth deployment and maintenance through automated CI/CD pipelines.

3. Key Modules

a) User Authentication and Authorization

We'll use Firebase Authentication with Multi-Factor Authentication (MFA) to guarantee that only authorized users can access the system. Role-Based Access Control (RBAC) will be implemented to assign roles like Admin, Doctor, and Patient, ensuring data access is tailored to user permissions.

b) Frontend Interface

Using the React Framework, we'll build a modular, component-based user interface that enhances user interaction. TypeScript will be utilized to enforce static typing, which boosts code reliability and maintainability. Tailwind CSS will help us develop the UI quickly with its utility-first styling approach.

c) Data Storage and Security

For secure storage of medical records, images, and documents, we'll rely on Firebase Cloud Storage. End-to-End Encryption will be in place to protect data both at rest and in transit. Firebase Security Rules will be applied to enforce granular access controls, preventing unauthorized data access.

d) Data Management and Backup

We'll use Fire store Database to store structured metadata and user-specific records. Data Versioning will help us track changes in healthcare reports, preventing accidental modifications. Automated Backups will ensure regular backups of Fire store data and storage content, minimizing the risk of data loss.

VII. ARCHITECTURE DIAGRAM

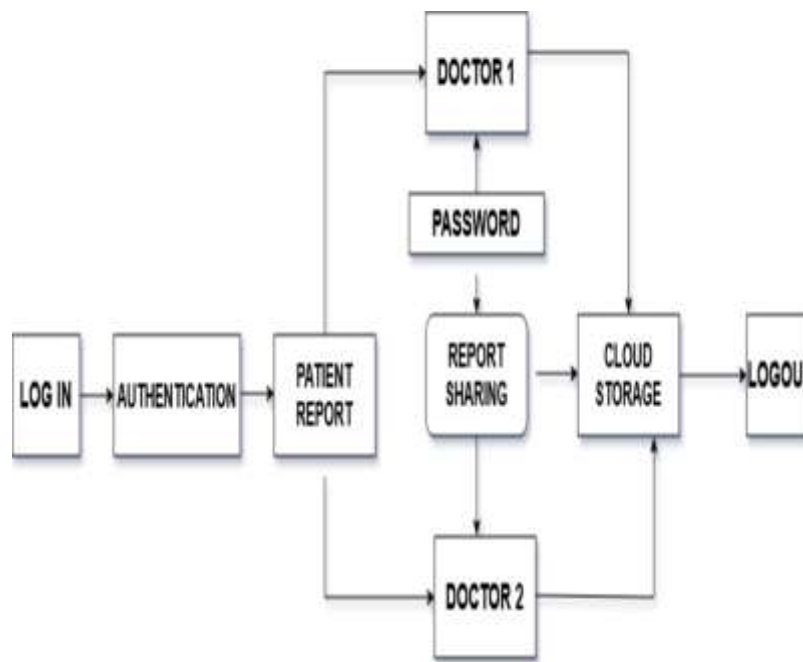


Fig 1: Architecture

VIII. METHODOLOGY

The methodology lays out a clear and organized approach for creating, developing, and launching a secure and scalable healthcare data management platform. It's broken down into several phases, each concentrating on essential elements like system architecture, implementation, security measures, and deployment.

1. Requirement Analysis and System Design

We kick things off by pinpointing the key functional and non-functional requirements through interviews with stakeholders, regulatory assessments, and feasibility studies. Next, we define the system goals, which include secure data storage, smooth user authentication, and scalable deployment. We design a three-tier architecture that includes: Frontend Layer: Built with React components using TypeScript and styled with Tailwind CSS. Backend and Storage Layer: Utilizing Firebase Authentication for secure user management and Firebase Cloud Storage for storing healthcare data. Deployment and Monitoring Layer: Implementing CI/CD pipelines that work seamlessly with GitHub Actions and Firebase Hosting. We establish security protocols that cover: Role-Based Access Control (RBAC): Assigning user roles to manage data access effectively.

End-to-End Encryption: Ensuring data remains secure during both transmission and storage. Audit Logging: Setting up logging mechanisms to monitor data access and modifications.

2. Frontend Development

We focus on creating a responsive and user-friendly interface using React and Tailwind CSS. By developing reusable components, we ensure a consistent design while minimizing development effort. We also use TypeScript to boost code quality and cut down on runtime errors. We take advantage of Vite for quick development cycles and Hot Module Replacement (HMR) to see changes in real-time. Additionally, we optimize frontend assets and reduce build size to enhance application performance.

3. Authentication and Data Storage

Set up Firebase Authentication to allow various methods for users to log in:

- Email/Password
- Phone Authentication
- OAuth Providers (like Google and others)

Establish roles such as Admin, Doctor, and Patient, each with their own data access levels. Utilize Firebase Security Rules to enforce detailed access control, ensuring that unauthorized users are kept out. Leverage Firebase Cloud Storage for keeping healthcare reports, images, and sensitive documents safe. Set up bucket-level security policies to block unauthorized access. Use Fire store Database to manage user metadata and role-based configurations.

4. Security Implementation and Compliance

Implement MFA to add an extra layer of security for user accounts, helping to reduce the risk of unauthorized access. Make sure all data at rest is encrypted using AES-256 encryption. Guarantee secure data transmission with TLS (Transport Layer Security). Create detailed security rules to safeguard data stored in Firebase Cloud Storage. Set up audit logging to monitor and record all access, changes, and retrievals of healthcare data. Perform regular security audits to ensure alignment with HIPAA and other regulatory standards. Establish controlled access, maintain audit trails, and use data encryption to meet compliance requirements.

5. Backend and API Development

Craft a normalized schema to effectively manage user roles, patient records, and access permissions. Enable automatic data synchronization for real-time access. Create RESTful APIs to facilitate data communication between the frontend and Firebase. Secure these APIs with token-based authentication and HTTPS.

6. Continuous Integration and Deployment (CI/CD)

Utilize GitHub for version control and to foster collaborative development. Set up GitHub Actions to automate the processes of testing, building, and deployment. Streamline your deployment processes with Firebase CLI to guarantee smooth updates. Establish production and staging environments to effectively manage various deployment states.

7. Deployment and Hosting Strategy

Launch the frontend on Firebase Hosting, which offers automatic SSL encryption and global CDN distribution, enhancing both performance and security. Host your Fire store database and Cloud Storage with real-time data synchronization for seamless access. Take advantage of Firebase's auto-scaling features to efficiently manage high traffic and large volumes of data.

8. Data Backup and Recovery

Set up regular backups for Fire store data and Cloud Storage to ensure data redundancy. Schedule periodic snapshots to enable rollback in the event of data corruption. Keep version control for sensitive documents to monitor changes and avoid accidental data loss.

9. Testing and Quality Assurance

Carry out unit tests for React components and APIs. Perform integration tests to confirm that system interactions are functioning correctly. Uncover potential vulnerabilities through penetration testing and security audits. Ensure that RBAC policies and MFA implementation are validated. Engage stakeholders in UAT to confirm that the system's functionality aligns with business requirements.

10. Monitoring and Maintenance

Set up real-time monitoring tools to keep an eye on how your application is doing and performing. Make sure to configure error reporting and alert systems so you can tackle any issues as soon as they pop up. Stay on top of your security policies by regularly updating them and conducting audits to ensure you're still in line with regulations. Don't forget to roll out bug fixes and feature enhancements based on what users are saying.

IX. RESULT and DISCUSSION

Using Firebase Authentication the system encrypts the username and password for users and verifies who is accessing the patient data. Once authenticated a doctor can view Patient Reports which are stored in a secure environment. The reports are shared between doctors by a Password protected mechanism which makes sure only authorized people have access to sensitive information. Firebase Cloud Storage securely stores patient reports. It enables doctors to easily share reports. Sharing reports improves health care and doctor efficiency. End-to-End Encryption ensures integrity of the data. It locks the data during transmission. This guarantees the integrity of the report. You can access the report only from authenticated doctors. Session Management tracks the user activity.

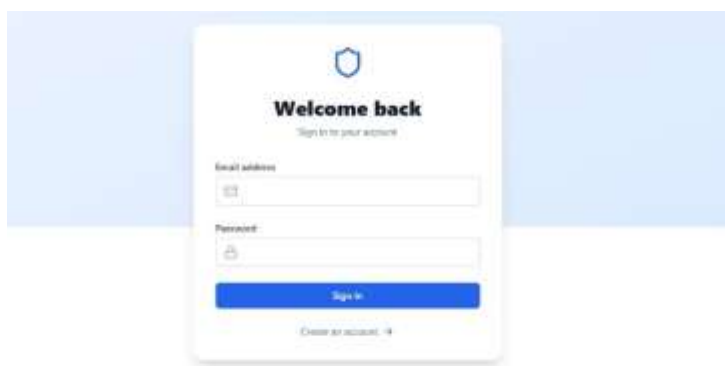
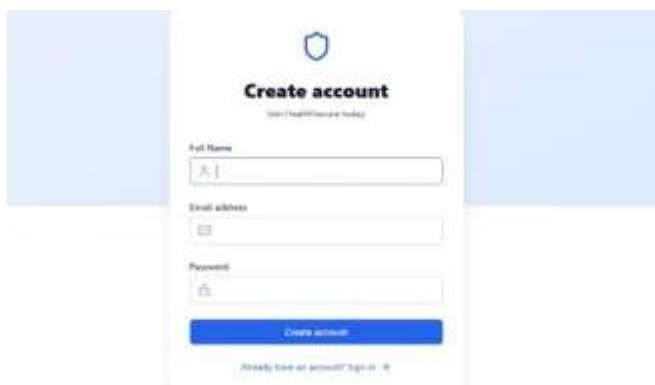
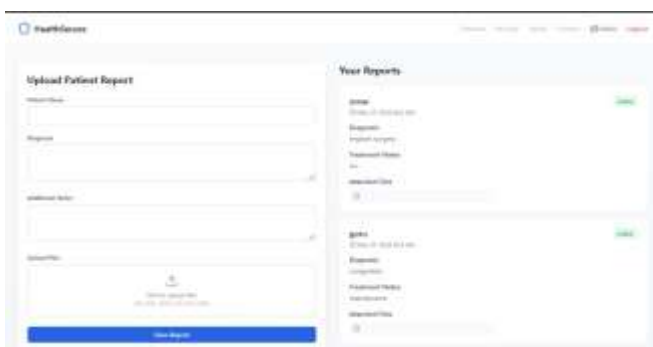


Fig 2: Login Page



The image shows a 'Create account' form for HealthSecure. At the top, there is a shield icon and the text 'Create account' with a subtext '(Join HealthSecure today)'. Below this, there are three input fields: 'Full Name' with a person icon, 'Email address' with an envelope icon, and 'Password' with a lock icon. A blue 'Create account' button is positioned below the password field. At the bottom, there is a link that says '(Already have an account? Sign in)'.

Fig 3: Create Account



The image shows a 'Patient Report' form in the HealthSecure interface. On the left, there is a section titled 'Upload Patient Report' with fields for 'Report ID', 'Report', 'Patient Name', and 'Patient ID'. Below these fields is a 'Upload Report' button. On the right, there is a section titled 'Your Reports' which displays a list of reports. Each report entry includes a status (e.g., 'Completed'), a date (e.g., '2024-03-20 10:30 AM'), and a list of details (e.g., 'Diagnosis', 'Treatment Plan', 'Prescription').

Fig 4: Patient Report



The image shows a 'Share Patient Report' modal window. At the top, it says 'Share Patient Report with Dr. James Wilson'. Below this, a green box contains the text 'Report Shared Successfully!'. Underneath, it displays 'Access Code: OZVOPNGQ'. A message below the code states: 'This code will expire in 24 hours. Please share this code securely with Dr. James Wilson.' At the bottom, there is a 'Close' button.

Fig 5: Final Report Sample

X. CONCLUSION AND FUTURE DIRECTIONS

The introduced framework effectively combines blockchain technology, encryption methods, and secure communication protocols to tighten the security and privacy of cloud-based health systems. The framework tackles key challenges including centralization threats, privacy threats, as well as compliance demands without compromising system performance. For the future it might consider using higher AI techniques to detect anomalies and preventing authorized access. Furthermore there could be a place for blockchain technology in improving data integrity and transparency.

XI. REFERENCES

- [1] Zhang, Y., & Li, X. (2023). A comprehensive review of privacy-preserving techniques for cloud-based healthcare systems. *IEEE Transactions on Cloud Computing*, 11(2), 345-360.
- [2] Chen, M., Wang, J., & Liu, H. (2023). Secure and scalable data sharing in cloud healthcare: A blockchain and homomorphic encryption approach. *Journal of Medical Systems*, 47(1), 12-25.
- [3] Patel, R., & Shah, P. (2023). Federated learning for privacy-preserving medical data analysis: Challenges and future directions. *Artificial Intelligence in Medicine*, 135, 102412.
- [4] Kumar, S., & Agarwal, V. (2023). Quantum-safe cryptographic solutions for healthcare data security. *Journal of Cybersecurity and Privacy*, 6(3), 87-102.
- [5] Lee, C., & Park, J. (2023). Enhancing interoperability and security in cloud healthcare through AI-driven access control. *ACM Transactions on Internet Technology*, 23(4), 56-72.
- [6] Zhao, H., & Lin, W. (2023). Blockchain-enabled secure data sharing framework for cloud healthcare systems. *IEEE Internet of Things Journal*, 10(2), 1563-1577.
- [7] Wu, T., & Sun, L. (2023). Privacy-preserving machine learning in healthcare: A survey on federated learning and differential privacy techniques. *Healthcare Informatics Research*, 29(1), 45-61.
- [8] Gupta, R., & Mehta, D. (2023). Homomorphic encryption for secure computation in cloud-based electronic health records. *Journal of Biomedical Informatics*, 139, 104453.
- [9] Yang, B., & He, Z. (2023). AI-driven anomaly detection for privacy risk mitigation in cloud-based healthcare environments. *Future Generation Computer Systems*, 152, 117.
- [10] Pandey, P., Voorsluys, W., Niu, S., Khandoker, A., & Buyya, R. (2021). An Internet of Things framework for the healthcare system: use cases, challenges, and blockchain solutions. *International Journal of Networked and Distributed Computing*, 9(3), 174–192.
- [11] Kuo, T. T., Kim, H. E., & Ohno-Machado, L. (2019). Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association*, 24(6), 1211–1220.
- [12] Wu, F., Wu, T., Yuce, M. R., & Lu, J. C. (2018). Protecting personal health information in Internet of Things systems: challenges, current solutions, and future opportunities. *IEEE Access*, 6, 3660–3671.
- [13] Kaur, H., Alam, M. A., Jameel, R., Mourya, A. K., & Chang, V. (2020). A proposed healthcare system using blockchain technology. *Healthcare*, 8(4), 461.
- [14] Jiang, S., Cao, J., Wu, H., Yang, Y., Ma, M., & He, J. (2020). BC-PPS: A blockchain-based privacy-preserving data sharing for electronic medical records. *IEEE Access*, 8, 185487–185498.
- [15] Hardjono, T., & Smith, N. (2021). Cloud-based commissioning of constrained IoT devices using permissioned blockchains. *IEEE Access*, 9, 107674–107.