PRIVACY PRESERVATION OF ELECTRONIC HEALTH RECORDS USING BLOCKCHAIN TECHNOLOGY: HEALTHCHAIN

by

SHEKHA CHENTHARA

A Dissertation Submitted in Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Institute for Sustainable Industries and Liveable Cities (ISILC)

VU Research

Victoria University, Melbourne, Australia

ABSTRACT

PRIVACY PRESERVATION OF ELECTRONIC HEALTH RECORDS USING BLOCKCHAIN TECHNOLOGY: HEALTHCHAIN

Shekha Chenthara, Ph.D.

Victoria University, 2021

The right to privacy is the most fundamental right of a citizen in any country. Electronic Health Records (EHRs) in healthcare has faced problems with privacy breaches, insider outsider attacks and unauthenticated record access in recent years, the most serious being related to the privacy and security of medical data. Ensuring privacy and security while handling patient data is of the utmost importance as a patient's information should only be released to others with the patient's permission or if it is allowed by law.

Electronic health data (EHD) is an emerging health information exchange model that enables healthcare providers and patients to efficiently store and share their private healthcare information from any place and at any time as required. Generally, *cloud* services provide the infrastructure by reducing the cost of storing, processing and updating information with improved efficiency and quality. However, the privacy of EHRs is a significant hurdle when outsourcing private health data in the cloud because there is a higher risk of health information being leaked to unauthorized parties. Several existing techniques can analyse the security and privacy issues associated with e-healthcare services. These methods are designed for single databases, or databases with an authentication centre and thus cannot adequately protect the data from insider attacks. In fact, storing EHRs on centralized databases increases the security risk footprint and requires trust in a single authority. Therefore, this research study mainly focuses on how to ensure patient privacy and security while sharing sensitive data between the same or different organisations as well as healthcare providers in a distributed environment.

This research successfully proposes and implements a permissioned blockchain

framework named Healthchain, which maintains the security, privacy, scalability and integrity of the e-health data. The blockchain is built on Hyperledger Fabric, a permissioned distributed ledger solution by employing Hyperledger Composer and stores EHRs by utilizing InterPlanetary File System (IPFS) to build the decentralized web applications. Healthchain builds a two-pronged solution (i) an on-chain solution implemented on the secure network of Hyperledger Fabric which utilizes the state database Couch DB, (ii) an off-chain solution to securely store encrypted data via IPFS. The Healthchain architecture employs Practical Byzantine Fault Tolerance (PBFT) as the distributed network consensus processes to determine which block is to be added to the blockchain. Healthchain Hyperledger Fabric leverages container technology to host smart contracts called "chaincode" that comprises the application logic of this system. This research aimed at contributing towards the scalability in blockchain by storing the data hashes of health records on chain and the actual data is stored cryptographically off chain in IPFS, the decentralized storage. Moreover, the data stored in the IPFS will be encrypted by using special public key cryptographic algorithms to create robust blockchain solutions for EHD.

This research study develops a privacy preserving framework with three main core contributions to the e-Health ecosystem: *(i) it contributes a privacy preserving patient-centric framework namely Healthchain; (ii) introduces an efficient referral mechanism for the effective sharing of healthcare records; and (iii) prevents prescription drug abuse by performing drug tracking transactions employing smart contract functionality to create a smart health care ecosystem.* The results demonstrates that the developed prototype ensures that healthcare records are not traceable to illegal disclosure as the model only stores the encrypted hash of records and is proven to be effective in terms of enhanced data privacy, data security, improved data scalability, interoperability and data integrity when accessing and sharing medical records among stakeholders across the Healthchain network. This research develops a foolproof security solution against cyber-attacks by exploiting the inherent features of the blockchain, thereby contributing to the robustness of healthcare information sharing systems and also unravels the potential for blockchain in health IT solutions.

DECLARATION OF AUTHENTICITY

I, Shekha Chenthara, declare that the PhD thesis titled "Privacy Preservation Of Electronic Health Records Using Blockchain Technology: HEALTHCHAIN" is no more than 100,000 words in length including quotes and exclusive of tables, figures, appendix, footnotes and references. The contents of this thesis, in whole or part, have not been submitted previously for the award of any other academic degree or diploma. Except otherwise mentioned, this thesis is my own work.

SHEKHA CHENTHARA 17/03/2021

ACKNOWLEDGMENTS

Throughout the completion of this research, I have received a great deal of support, motivation, guidance, and assistance from the following people.

First and foremost, I would like to thank my principal supervisor, Professor Hua Wang, for being a wonderful supervisor whose expertise was invaluable in formulating the research motivation and for his guidance throughout the research. Thank you for your continued support, kindness, patience, and encouragement throughout my PhD candidature. I would not have been able to submit the research papers in Q1 journals and quality conferences without your support. I would like to express my endless gratitude to you for your professional and personal advice and for being there during my happiness, sorrows, failures, and success. Thank you for honouring my successes and inspiring me to do better after any setback. From the bottom of my heart, I thank you very much for making me what I am today.

I deeply acknowledge my associate supervisor, Dr. Khandakar Ahmed for his continual encouragement and guidance throughout my research study. He is a remarkable teacher and a person, who gave critical feedback and comments at every stage of this research. Without his guidance and mentoring, it would have been difficult for me to complete this research. Thank you so much for being there as a teacher, friend, a pillar of support and well-wisher to me. Thank you so much for advising me to apply for the 2019 SummerTech LIVE Internship organized by the Victorian Government and for supporting my participation and guiding the presentation of the work. Thanks again for suggesting that I demonstrate the research work on the VU Open Day 2019 which also turned out to be a wonderful experience. Thank you for believing in me and giving me teaching assistance and helping me to rectify my errors and approach in the academic field. I am highly grateful to Dr. Khandakar for his timely advice, research contribution, formulating the technical aspects, for developing and building the proof of concept at every phase of my research.

It is a genuine pleasure to express my thanks and gratitude to my industrial partner

Dr Frank Whittaker for being a mentor and guide throughout this research. Thank you for giving me the opportunity to be a part of the ARC Linkage research program. Thank you for introducing me to various aspects of research, for the many critical research discussions, and for the inspiration, dynamism and encouragement that you offered throughout my research candidature.

I am especially grateful to the Director of the Centre for Applied Informatics, Professor Yanchun Zhang, for his guidance, motivation, and support during the team meetings and research group discussions. I would also like to extend my sincere appreciation to Professor Yuan Miao, Head of Information Technology for his invaluable advice and guidance throughout the research. The stimulating discussions we had proved instrumental during various stages of research and especially when I was having trouble finding research ideas. Thank you Prof. Yuan for giving me a chance to host the online "International Conference on Information Visualisation 2020", which helped me gain much confidence to engage and associate with various professionals from different parts of the world. Thank you very much Professor for the encouragement and helping me to tackle difficult situations in research.

Thank you to Prof. Randall Robinson, Director of the Institute for Sustainable Industries and Liveable Cities at Victoria University for the support and assistance needed for this research study. The research presented in this thesis was supported by an Australian Research Council Linkage Scholarship, offered by Victoria University and industrial collaboration with Nexus e-care. This financial support is gratefully acknowledged. I would like to express my gratitude to deputy director Dianne Hall, Elizabeth Smith and Dr Lesley Birch for their support and guidance. I would also like to thank the Research Scholarships and Funding team at Victoria University who supported me by providing research funding during the COVID-19 pandemic. I would also like to acknowledge the technical, academic and financial support given by ISILC staff including teaching, administrative and facilities staff during my research study.

Thank you very much to my former colleague, supervisor and mentor Dr Sudheep Elayidom, Cochin University of Science and Technology, India for introducing me to the field of research and giving me an opportunity to explore research before I commenced

my PhD in Australia. Thank you so much to my friend Lija Mohan for her support and encouragement during my research.

I am forever grateful to my friends and colleagues for their inspiring discussions, especially Sarathkumar Rangarajan for his endless motivation, Ujjwal Datta, Rubina Sarki, and Douglas Pinto Sampaio Gomes for their support throughout the research. My appreciation also goes to Sonal Chanana and Taman Shergill for their unwavering support and belief in me when I needed it the most.

My deepest appreciation goes to my brother Shameer Chenthara who was always there from the first day to the final day of my PhD candidature with daily motivation, encouragement, support and love. Thank you so much to my spouse Nabil Varis who gave me continued support and undertook the domestic responsibilities while I was working. Thank you very much to my parents for their constant encouragement and for taking care of my daughter for a period when I came to Australia to pursue my research studies. Last but not the least, thanks to my 7yr old daughter Suhana Varis for all her love and patience while I was completing my PhD research.

This study was supported by an Australian Research Council Linkage Scholarship.

DEDICATION

This thesis is dedicated to my brother for his endless motivation, my parents for their constant encouragement, my spouse for his continued support and to my beloved daughter for her love throughout the research.

Contents

List of Figures

List of Tables

LIST OF PUBLICATIONS

The following articles are published or submitted in International Journals and Conference based on this research work.

Journal articles:

1. **Shekha Chenthara**, Khandakar Ahmed, Hua Wang, Frank Whittaker, Zhenxiang Chen (2020), "Healthchain:A Novel Framework on Privacy Preservation of Electronic Health Records using Blockchain Technology", PLOS ONE Journal, December 9, 2020.
2. **Shekha Chenthara**, Khandakar Ahmed, Hua Wang, Frank Whittaker (2019), "Security and Privacy Preserving Challenges of e-Health Solutions in Cloud Computing," IEEE access, vol.7, pp. 74361-74382, 2019.
3. **Shekha Chenthara**, Khandakar Ahmed, Frank Whittaker, "Privacy-Preserving Data Sharing using Multi-layer Access Control Model in Electronic Health Environment," EAI Endorsed Transactions on Scalable Information Systems, vol.6, no.22, 2019.

Conferences:

1. **Shekha Chenthara**, Khandakar Ahmed, Hua Wang and Frank Whittaker, "A Novel Blockchain Based Smart Contract System for eReferral in Healthcare: HealthChain", International Conference on Health Information Science, HIS2020, pp.91-102, 2020.
2. **Shekha Chenthara**, Khandakar Ahmed, Hua Wang, Frank Whittaker and Ke Ji, "A Blockchain Based Model for Curbing Doctors Shopping and Ensuring Provenance Management" 2020 International Conference on Networking and Network Applications (NaNA), IEEE, pp. 186-192, 2021.

Book Chapters:

1. **Shekha Chenthara**, Hua Wang, Khandakar Ahmed, Frank Whittaker, "Security and Privacy in Big Data Environment," Book Chapter published in Encyclopedia of Big data Technologies, Springer, 2018.

Other Presentations

1. Research Presentation during SummerTech Live Program 2019.
2. Research Presentation during VU Open Day 2019.

Chapter 1

Introduction

1.1 Introduction

With the increasing use of big data across multiple domains viz science, engineering, commercial fields and so on, it has become an area of research interest as there is growing concern over big data security and the privacy of individuals in every sector [37]. We come across data in every possible form, through social media sites, sensor networks, digital images or videos, cell phones, Global Positioning System (GPS) signals, purchase transaction records, weblogs, medical records, archives, military surveillance, e-commerce, complex scientific research and numerous other fields and the volume of data amounts to some quintillion bytes of data generated daily. This data is what we call big data [144].

The evolution of big data has enabled the healthcare industry to transform health data to electronic health data (EHD) or electronic health records(EHRs). EHD includes electronic or computerised patient records including demographics, medical histories, medication and allergies, immunisation status, laboratory test results, radiology images, billing information and so on. The advantages of EHRs include easier and swift clinical data access, ability to maintain effective clinical workflows, mitigation of medical errors, enhanced patient safety, reduced medical costs and better and stronger support

for clinical decision-making. Realising the benefits offered by EHD systems, more than 90% of healthcare institutions in Australia and across the globe have adopted this system to facilitate effective medical resource allocation and efficient healthcare [75]. EHRs have been also widely used to enable healthcare providers and patients to create, store, manage and access healthcare information on demand from any place and at any time. Generally, cloud services provide the best infrastructure by reducing the cost of storing, processing, and updating information with improved efficiency and quality.

Cloud computing is an evolving paradigm in digital technology and is being extensively used in the healthcare industry [59]. The large-scale proliferation of health information in the age of big data necessitates the burgeoning role of cloud networks not only for hosting unlimited amounts of data but also to facilitate the easy exchange or transmission of medical data among various stakeholders [88]. It facilitates the creation, storage and retrieval of healthcare information by all stakeholders viz healthcare providers, doctors and patients with ease irrespective of the barriers posed by time and space. Since the data is running on a wide network of remote servers, which are integrated and operated as a single ecosystem accessed from different locations by multiple users, it is susceptible to intrusion or compromise, thereby posing a threat to privacy and security. Moreover, as the majority of medical data is highly sensitive and strictly confidential, its storage on third party servers naturally increases these vulnerabilities [1]. Undoubtedly, the most challenging and concerning problem is security and privacy. Many studies show that big data will harm the users' privacy if it is not properly handled[102]. The security and privacy issues which should be of concern in the big data context include the following: (1) The personal information of a person when combined with external large data sets leads to the inference of new facts about that person whereby these facts about the person are sometimes secret and the person might not want the data owner to know or any person to know about them; (2) Information regarding the users (people) is collected and exploited to add value to the business of any organization. This is done by creating insights into their lives which they are unaware of; (3) Another consequence is social stratification where a literate person can take advantage of big data predictive analysis whereas the illiterate/ underprivileged will be worse off, as is evident in developing countries where the digital divide is very

much prevalent; (4) If used by law enforcement agencies, big data will increase the chance of certain tagged people suffering from adverse consequences without the ability to defend themselves nor having the knowledge that they are being discriminated against [70]. In light of this and the susceptible nature of health information in the public domain, there is an imminent need to devise a more secure, efficient and effective mechanism for sharing and accessing data among stakeholders [36].

Storing this confidential data in the cloud, can often be a major obstacle for the efficient utilization and processing of patient data. Since most EHD is sensitive and strictly confidential, the security of stored medical data is a major concern. This research aims to build a task-based framework that effectively and securely shares patient data between different organisations and stakeholders whilst preserving patient confidentiality. Blockchain is one of the approaches to address most of the shortcomings of the current distributed framework by implementing a patient-centered electronic healthcare system, namely Patient Controlled Electronic Health Record System (PCEHR), in which the patient is the sole consent provider of their data to all stakeholders except in emergency situations.

1.2 Motivation

With the proliferation of big data, a common solution is outsourcing large volumes of data into third party cloud storage which poses the threat of breach or data leakage. Privacy and security of data is a major hurdle when outsourcing private data in third-party cloud servers, as there is a possibility of leaking or sharing sensitive information with unauthorized entities. So, to ensure its legitimate and authorized usage, security is paramount so that the right person gets the right data at the right time in the right way. Identity and access management in healthcare is of prime importance in this regard, and this should be complemented by periodic audit trails to detect a range of events from users logging on to the system, acquiring authentication and accessing files or records and executing applications as authorized. The next most important motivation is to safeguard

identity provisioning, which involves the secure and timely management of provisioning and deprovisioning of users in the cloud. There is also a high need to manage the various stakeholders' access to the cloud environment in an effective manner. Thus, this work incorporates three main factors viz identification, authentication and authorization in the e-health domain.

Another main motivation for undertaking this research is to discuss various existing security and privacy preserving mechanisms in the healthcare environment, their strengths and drawbacks that makes EHRs vulnerable to threats in the cloud arena and to devise a foolproof mechanism to address this. The available privacy preserving mechanisms are inadequate to ensure foolproof security for the seemly management of EHRs in the cloud. E-health data contains diverse sensitive and confidential information ranging from patient data to financial information such as social security numbers, and credit card details, whose leakage not only throws open patients' sensitive information and causes financial losses but also infringes on the most fundamental right of a citizen in any country i.e. the right to privacy. The main issue faced by health records in cloud servers is internal attackers who have authorized credentials to access data within an organization in which the database administrator or key manager is the attacker which is significantly worse than external attacks. Another major threat is the openness of data to cloud providers which poses the dangers of data threats or misuse. A lack of interoperability in EHRs is one of the main issues faced by the healthcare industry today. Health data in prevalent systems is fragmented and is challenging to share with healthcare providers or stakeholders due to their varying formats and standards. This means that it is difficult to aggregate and examine patient data which prevents the efficacy of EHR sharing in emergency situations. Another major drawback is that since healthcare records are stored in centralized databases in silos, healthcare data becomes an extremely tempting target for attackers. Several research studies show that centralization increases the security risk and requires trust in a single authority. Moreover, in the existing system, patients are not in complete control of their health records since these are managed by service providers. The centralized databases can leave us vulnerable to attacks that escalate cyber threats, from the recent Ransomware attack to the Equifax attack which hinders the privacy and security of EHRs. Despite the

outstanding features the existing healthcare industry provides, it fails to provide an efficient way to store, share and analyze health data in a globally unified way. For example, earlier this year, hackers broke into the databases of Community Health Systems (CHS), one of the largest hospital groups in the United States and accessed personal health information, names, addresses and personal data including social security numbers from around 4.5 million patients. Hackers from the Internet vigilante group Anonymous also targeted several hospitals, launching a DDoS attack on the hospital website as an act of “hacktivism" [2]. Therefore, there is an urgent need to protect the privacy, security, confidentiality, integrity, and availability of sensitive information pertaining to individuals’ data in general. In this context, cybersecurity is required to prevent, detect, and act on unauthorized access to a health system and its information. However, several issues such as data encryption, secure storage, strong authentication, access control, key management, and efficient user revocation are yet to be addressed and resolved. This scenario has motivated us to devise a new mechanism which offers better safety and security measures in the e-healthcare infrastructure.

Most of the aforementioned problems will be resolved by employing Hyperledger Fabric as the underlying permissioned blockchain technology and IPFS as the decentralised file system for secure data storage for the e-health environment that provides efficient and secure sharing of health records in the e-health ecosystem. Blockchain offers interoperability, scalability, data integrity, data privacy and security provided by its secure hash algorithm and consensus property. In this research work, we propose a permissioned blockchain framework based on Hyperledger Fabric as the underlying structure and design a working prototype which can be used for efficient data sharing, the management of health records and systematic access control. Consequently, this research introduces a permissioned patient-centric blockchain namely Healthchain, for EHRs which eliminates most of the bottlenecks and evades the likelihood of a single point of failure in the existing systems. The interoperability challenges in healthcare are resolved by the Healthchain framework in the way it is built. i.e. the Healthchain framework stores the patients’ history by syncing records in different formats by accessing data via the REST server API by employing self-governing and constantly executing smart contracts in the framework. Also, the patient has complete control over

their healthcare records by providing access and identity permissions to authorized stakeholders. Moreover, the immutability of health records is also achieved by cryptographically storing the data inside i.e. by storing the hash values of data in the blockchain and storing encrypted healthcare records in the offchain IPFS database which makes the framework tamper resistant. Healthchain is a decentralised framework and is built in such a way that nobody can tamper with the records as the data transactions are linked and a consensus of stakeholders needs to agree to add data in the network. Our system contributes to healthcare by addressing most of the challenges associated with data privacy, security, interoperability, scalability, trust, immutability and data integrity.

1.3 Research Problems

The healthcare industry has been facing problems with privacy breaches and unauthenticated record access as the data is stored in third-party cloud servers where the user doesn't have direct control. Patient privacy is paramount in healthcare organisations including hospitals, medical centres, independent physician groups and insurance providers. The main aspect is the right of an individual to ensure their information is not disclosed to others, to be left alone from surveillance or interference from other individuals, organizations or the government. Their data need to be used for practical purposes in an effective manner, and data security and patient privacy should be ensured.

Therefore, this research focuses on reviewing the taxonomy of EHR privacy preserving mechanisms in the *cloud*, studying different blockchain technologies to identify relevant strategies to strengthen or revamp the existing security infrastructure by introducing the features of blockchain technology as a protection mechanism in e-health. This research work also envisages welfare orientation, i.e. to contribute to the patients and to society in general. The following research challenges were identified.

RQ1: How to employ an efficient access control mechanism and encryption technique

using blockchain technology in e-health data storage to address the shortcomings of the existing systems?

RQ2: How can a new framework be designed, developed and analysed to achieve a proof of concept (POC) to demonstrate patient privacy and data security?

Sub RQ1: To what extent can cyber security issues be resolved by employing a privacy-preserving framework to maintain the integrity of the EHD?

RQ3: How can smart contracts be employed for distributed ledger technologies to effectively share and transfer medical records among the stakeholders?

RQ4: How can IPFS be devised to facilitate data scalability and data security in the Healthchain framework?

1.4 Research Aims

The overall aim of the research is to develop a novel task-based framework for effective data sharing on EHD database federations while protecting data against both outsider and insider attacks, providing visualised, dynamic support to medical staff and government resource planners and policymakers. The research aim presented in this thesis are as follows:

• To implement a privacy preserved healthcare system, this research builds a permissioned blockchain framework namely Healthchain aiming to achieve patient privacy and security by providing efficient access control mechanisms and encryption techniques to build the decentralized web application.

• The proposed Healthchain framework builds a Distributed Ledger Technology(DLT) chaincodes known as smart contracts that comprise the application logic of the framework to ensure the efficient transfer and sharing of health data among stakeholders.

• To build secure data storage, the data stored in the IPFS will be encrypted using PKI cryptographic algorithms to create robust blockchain solutions for EHD.
• The proposed Healthchain framework builds provenance data by keeping patients' entire medical histories in a blockchain wallet to ensure the integrity of health records.
• The proposed system also includes a drug supply chain smart contract management system by performing drug tracking transactions on a blockchain that resolves prescription abuse or doctor shopping to create a smart health care ecosystem.
• This framework aims to build a scalable system by employing a decentralized data storage, IPFS.

This work develops a working prototype through which the blockchain approach is discussed and provides a foundation for developing security solutions against cyber-attacks by exploiting the inherent features of the blockchain, and thus contributes to the robustness of healthcare information sharing environments.

1.5 Objectives

The main objective of this research is to build a private blockchain-based system to share important and sensitive information which places the control of patient data in the patient's hands (Patient Centric) by employing a specific encryption mechanism to resolve the challenges related to secure storage and strong access control mechanisms to provide authorization to develop foolproof security solutions against cyber-attacks in a digital health environment. The individual objectives of this research are:-

• **Secure, immutable and decentralized EHR database with the patient owning her/his own health data.**

This research aims to develop a secure system for the efficient sharing of electronic health records among various stakeholders as well as healthcare organisations in a distributed environment. It also focusses on secure storage of health records while sharing and exchanging data among the stakeholders in a peer-to-peer network. This research builds immutability or tamper resistant healthcare records which is another

significant feature of this system. A decentralized database is proposed which eliminates centralisation that requires trust in a single authority. In addition, this work provides ownership and full control of their EHR to the patient.

• **Easy to share with selected or all EHRs as consented by the patient.**

This framework is proposed to develop in a way that patient is the sole entity to provide permissions to the stakeholders in the healthcare environment. Also, the patient can share selected records for a particular session to the permissioned users instead of sharing the entire patient details. The proposed framework allows patient to grant or deny access for EHR to the stakeholders based on the role and rule-based access control management rules. The access can be given to the entire EHR or to a composite view of the record based on the user permission. These rules will be stored in the blockchain and submitted to the blockchain channel through a transaction called business network transaction.

• **Full medical history of a patient at one single point.**

Another significant feature proposed by this work is its ability of provenance management to keep track of the health records in the user account. This system allows patient to keep track of the record history as well as the records that have been added or updated. The provenance or history of the patient record include the patients' accumulated data from clinical encounters and records of the types of data amassed such as vaccination histories, pathology reports, blood results, referral reports and are stored on the patient's Healthchain.

• **Easy verification of medical prescription.**

This research also contributes to improve the way opioids and prescriptions are administered and distributed by creating a secure framework for stakeholders in healthcare which prevents prescription drug abuse or doctors shopping. This can be done by recording all the transactions between clinician, patient and pharmacist by making it possible to determine the quantity of medication transferred, to whom the medicine was transferred, when it was transferred and the frequency of patient visits.

• **Increased transparency.**

This research proposes to develop a blockchain framework in which every transaction is transparent to the users in the network. This work builds a decentralised structure and

can provide an immutable and timestamped log of records in which all transactions are transparent to stakeholders in the permissioned blockchain network. Updates are made immediately available and transparent to all parties across the distributed database. Furthermore, this research builds in a way to store every transaction in the blockchain ledger, and all the participating peer nodes have ledger back up that promotes data transparency.

• **No insurance fraud.**

The first and most important advantage of blockchain is its trustworthiness. Claiming for treatment or services that haven't been delivered, using someone else's Medibank card, or giving fake information or papers are all examples of health insurance fraud. Fraud can be member fraud such as claiming a benefit that is not entitled or provider fraud such as claiming for services or products that weren't provided, invoicing for different items, charging members for something which is not clinically necessary etc. Since all the transactions between the stakeholders are transparent and trackable, this system can be designed to identify claim management process and only offer access to information needed for insurance claims.

1.6 Research Contributions

The research work carried out has achieved the aims and objective of this project. A working prototype based on blockchain technology has been implemented and evaluated using some healthcare use cases which integrates cryptographic components that incorporates solutions for a more secure and effective framework to store, transfer and access EHRs in the cloud environment. The private blockchain-based prototype namely Healthchain is a robust tamper-proof ledger as shown by the test results. The work builds a permissioned blockchain-based architecture called Healthchain by employing Hyperledger Fabric to securely and scalably share healthcare records to preserve patient privacy, deliver efficient permission management among stakeholders to enhance collaborative clinical decision support and comprehensive patient care. IPFS has been employed in this work to build a secure decentralised data storage. The prototype

designed is a user-centric model with a few stakeholders namely doctor, patient, receptionist and pharmacist that builds a permissioned Healthchain framework. The main contributions of this research are summarized as follows:

• A review and thematic classification of the literature is carried out. This research highlights a comprehensive classification of privacy preserving cryptographic and non-cryptographic approaches and their existing vulnerabilities and challenges in the e-health cloud that shows the issues that our Healthchain project has attempted to address in order to make the proposed solution more recognizable. In addition, this work also provides and describes key research areas from various aspects, including encryption methods, access control mechanisms as well as defining several key factors including the strengths and limitations of current techniques, and characterising each approach using several privacy preserving requirements such as IN (Integrity), CO (Condentiality), AU (Authenticity), NR (Non-represervation) AC (Accountability), AN (Anonymity) and UN (Unlinkability).

• This research builds a patient-centric Healthchain framework in which patients will have full control over their medical records, maintaining the security, privacy, scalability and integrity of e-health data. A permissioned blockchain namely Hyperledger Fabric, is employed to build the private Healthchain network. In addition, this work also utilizes a Rest Server, Hyperledger Composer for designing and modelling the blockchain business networks and to build smart contracts for the efficient functioning of the health data network.

• To maintain the efficiency and scalability of the blockchain network, this research proposes a decentralized storage viz IPFS that is used as an off-chain database for storing encrypted health records. Furthermore, the Healthchain framework employs CouchDB as the on-chain database which stores the unique cryptographic hash generated by IPFS. Because of its decentralized property, this framework ensures no single point of failure and also changes to the blockchain will be visible to the participants of the Healthchain network that are immutable.

• This research work proposes an effective PKI cryptographic algorithm for encrypting the data stored in the offchain database, IPFS to create robust blockchain solutions for

EHD.

• Our research design also proposes several access control rules and mechanisms to provide user access permissions to the authorized stakeholders and also does not involve any form of mining incentives beyond the efficient use of the system. This framework develops a working prototype in which the blockchain technique is analyzed and also unravels the possibility of blockchain in healthcare solutions.

• This work also proposes a Healthchain-based smart contracts algorithm for the effective sharing of healthcare records between clinicians and stakeholders in the healthcare industry.

1.7 Thesis Composition

This research study includes a literature review, a description of the study steps, a discussion of the algorithm design and simulations, an evaluation of the outcomes of simulations, a comparison of the findings with alternatives contained in the literature and defining work for the future. The organization of the thesis is summarized as follows:

Chapter 1 provides a research overview and presents the research aims, objectives, motivation, research problems and contributions. The introduction explains the reasons and motivations for conducting this research and why it is relevant and briefly explains the method followed.

Chapter 2 provides an extensive survey on security and privacy-preserving challenges of e-health solutions and various privacy-preserving approaches to ensure the privacy and security of EHRs in the cloud are presented. Currently the cloud is one of the pioneers in e-health data storage and data transfer among stakeholders. However, the problem is that EHRs are stored in centralized databases in silos. As a result, health data has become an extremely tempting target and prone to attacks. In addition, patients do not have complete control over their health records. This study discusses the majority of

approaches are incapable of withstanding internal and external attacks and fail to achieve security, privacy, interoperability and integrity of health data in the e-health cloud arena. The review of the cloud environment concludes with a solution to overcome the limitations in the existing system by introducing a patient-centered electronic health system using blockchain technology and possible research directions.

Chapter 3 introduces blockchain technology, its computational techniques in detail and compares several blockchain platforms including Ethereum, Quorum, Ganache, Hyperledger Fabric and distributed data storage such as Siacoin, Swarm, StorJ and IPFS. It also compares Ethereum healthchain and Hyperledger Fabric healthchain for the efficient storage of and access to healthcare records. Ethereum and Hyperledger Fabric technologies are at the forefront of the future medicare industry where data privacy, data security, scalability and data integrity are the dominant factors in the e-health environment. To demonstrate their comparative performance and efficiency several studies need to be employed. This chapter concludes with the solution that, since the healthcare field carries confidential and sensitive information, Hyperledger Fabric permissioned healthchain is a better solution and IPFS as the distributed storage as it can be the future Internet.

Chapter 4 details the development of the working prototype Healthchain which establishes a better, secure and transparent framework that maintains the privacy, security and integrity of EHRs. This model uses blockchain technology utilizing Hyperledger Fabric, Hyperledger Composer and decentralized storage, IPFS and the state database, Couch DB. This work also proposes an advanced public key encryption mechanism to effectively store data in the decentralized database. This work proposes a permissioned blockchain framework viz Hyperledger Fabric to be employed in the healthcare industry to keep EHRs tamper free and proposes IPFS to store the health data in a decentralized fashion. The framework is tested by utilizing access control mechanisms, smart contracts and IPFS as the decentralized storage for secure data storage. This chapter also present a provenance model to represent the provenance of the health records at any abstraction layer and present an abstract schema of the model. This model stores the data of a person from birth and via clinical encounters and

uploads data as a new block to their electronic health chain which contributes to the integrity of e-health data. This POC includes the successful implementation of a prototype which stores EHRs by securely maintaining the integrity, privacy, scalability and data security. This chapter addresses RQ1, RQ2 and Sub RQ1.

Chapter 5 This chapter introduces an efficient referral mechanism employing advanced smart contracts for the effective sharing of healthcare records between clinicians in the healthcare industry. This referral system is built on a patient-centric model and is limited to authorized providers in the healthdata network. This system is built by employing Hyperledger Fabric as the permissioned blockchain utilising Hyperledger Composer as the Rest Server which visualizes the couchDB and IPFS as decentralised data storage are combined for efficient and secure big data sharing in the healthcare sector. Furthermore, this work also conducts simulation studies to prove the scalability of IPFS as a decentralised file system. This chapter addresses RQ3.

Chapter 6 presents several access control models and smart contracts in Hyperledger Fabric for the effective management of pharmaceutical drug supply tracking in the healthcare industry. The pharmaceutical blockchain has the potential to improve the security, integrity, source of data and operation of effective medical supply chains in a transparent, unchanging and auditable way. This work proposes an efficient provenance mechanism using advanced smart contracts to effectively track and eliminate counterfeit medical drugs exchanging between stakeholders in the healthcare industry. RQ4 and RQ3 are discussed in this chapter.

Chapter 7 encapsulates the overall conclusions and analysis of the thesis and provides future research direction based on the studies in this thesis.

Chapter 2

Literature Review

2.1 Overview

The literature review offers context information that underpins the ongoing research and illustrates the state-of-the-art strategies, methods and approaches in the research subject field. A systematic and thorough analysis of security and privacy-preserving issues in e-health solutions showing different approaches to privacy-preserving electronic health records (EHRs) in the cloud is given. This study highlights the research challenges and directions regarding cyber protection to create a robust EHR safety model. This review also researches, examines and analyses various aspects of several journals including IEEE, Science Direct, Google Scholar, PubMed and ACM for papers on EHR approaches published between 2000 and 2018 and summarizes them in terms of the architecture types as well as evaluation strategies and discusses tasks such as security and privacy criteria for e-health data and EHR system architecture and various cryptographic and non-cryptographic approaches for EHR. This chapter surveys, investigates and reviews various aspects of several articles and identifies the following tasks:1) EHR security and privacy (2) security and privacy requirements of e-health data in the cloud (3) EHR cloud architecture and (4) diverse EHR cryptographic and non-cryptographic approaches and also discusses some crucial issues and the ample opportunities for advanced research related to the security and privacy of EHRs. This

chapter also offers a thorough analysis of cryptographic approaches such as Symmetric Key Encryption (SKE), Public Key Encryption (PKE), Attribute -Based Encryption (ABE), Searchable Symmetric Encryption (SSE), Proxy Re-encryption (PRE), Homomorphic Encryption and Non-cryptographic approaches which includes access control mechanisms such as Discretionary Access Control (DAC), Mandatary Access Control (MAC), Role-Based Access Control (RBAC), Rule-Based Access Control, Attribute-Based Access Control(ABAC), Identity-Based Access Control (IBAC) as well as their strengths and weaknesses. In addition, this review also researches a dual layer access control model named the Pseudo-Role Attribute based access control (PR-ABAC) mechanism or a multi layer access control (MLAC) mechanism that integrates attributes with roles for the secure sharing of EHR between multiple collaborators. Through this study, the review analyses the strengths, drawbacks, research problems of current privacy-preserving techniques and proposes a new model backed by blockchain technology, which can resolve some of the limitations and also provides a foolproof mechanism to preserve privacy and security efficiency in e-health data.

2.2 Security and Privacy Requirements of e-health Data in the Cloud

In this big data epoch, outsourcing health data to cloud servers poses the risk of several types of cyber attacks ranging from information disclosure, denial of service (DoS) attacks, man-in-the middle attacks to ransomware attacks which have greater ramifications beyond financial breaches or a loss of privacy [5]. Hence there is an imminent need to preserve and protect data to maintain patient confidentiality. The vital security and privacy requirements in e-health systems are: 1) Data integrity- ensures that the health information has not been altered by any unauthorised entity. 2) Data confidentiality- ensures that sensitive health data is prevented from reaching unauthorised individuals. Data encryption is the most substantial approach to ensure data confidentiality. 3) Authenticity- ensures that only authorised and authentic

authorities have access to sensitive health data. 4)Accountability- an obligation to be responsible and to justify the actions and decisions of individuals or organizations. 5) Audit- a requirement which ensures that health data is monitored and protected by keeping track of the activity log and assuring the users that their data is being kept private and secure. 6) Non-repudiation- refers to the non-denial of the authenticity of a sender and receiver. For instance, the patients or doctors can't repudiate after the embezzlement of health data 7) Anonymity- ensures that the identity of the subject is anonymous so cloud servers are unable to access the identity of the stored health data [36]. Cloud computing is a centralized mainframe computing paradigm owned by the cloud provider which is less patient-centric and is prone to insider attacks that makes the health records more vulnerable. This is one of the major downsides of cloud computing. Even though cloud techniques adhere to strict security measures, they do not offer a foolproof solution to be adopted into e-health, taking into account the security issues. Several innovative cloud protection strategies are discussed and some innovations are highlighted with their pros and cons in Table 2.1. Some of the advanced privacy-preserving frameworks can be implemented for e-health although others are not preferred due to security concerns, so they don not offer a foolproof solution for the e-health domain.

2.3 Overview of the E-health Systems in the Cloud

The e-health system is a recent healthcare innovation utilising electronic processes and communication. In an e-health system, EHR or EMR is a systematized aggregation of the electronic health information of patients [156]. These records involve all health data information including demographics, medical histories, medications, laboratory reports, radiology images, billing information and any additional sensitive patient information. The cloud offers a great service to both healthcare providers and patients in terms of cost-effective storage, and the processing and updating of information with enhanced efficiency and quality. Since all this data is stored in multiple servers, it can easily be accessed by users from various locations on demand. E-health systems promise rapid,

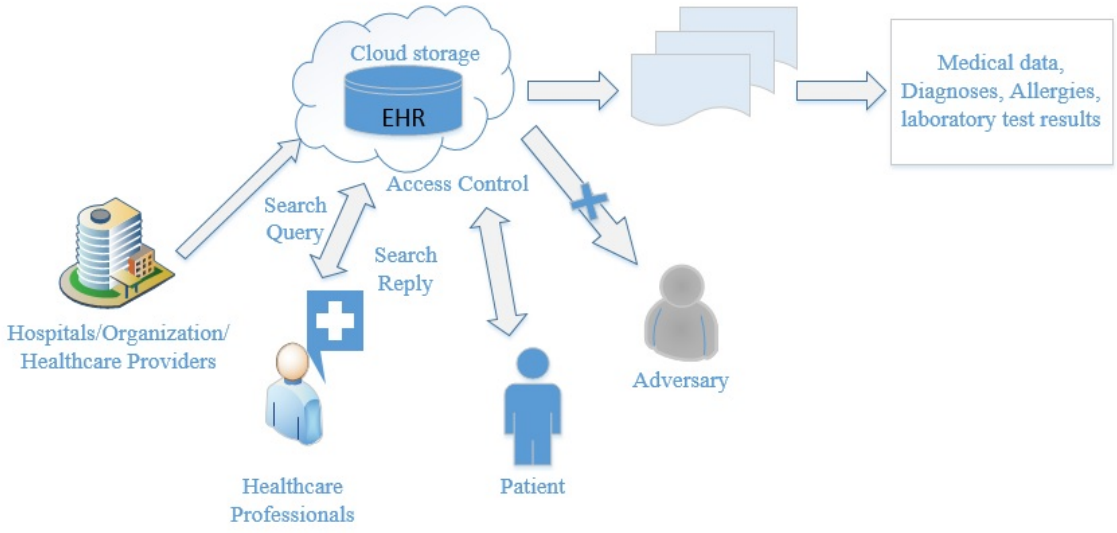


Fig. 2.1 Architecture of Electronic Health Data in the Cloud.

steadfast and on-demand access to medical records, fewer medical flaws and enhanced healthcare quality, however they equally expose patient privacy, via improper authorization and the misuse of EHR data. Therefore, security and privacy are considered critical requirements when sharing or accessing patient data between several stakeholders. An overview of the e-health architecture is depicted in Fig.2.1. E-health cloud architecture types can be public, private, hybrid or community according to the stored data. Since EHR data is strictly confidential, carries sensitive patient information is housed in third-party servers, access control mechanisms are required. Access control is a security barrier which preserves data privacy by restricting the operation and access of healthcare documents in the healthcare system. The predominant access control techniques in the healthcare systems are role-based access control (RBAC), attribute-based access control(ABAC) and identity-based access control (IBAC) techniques. Role-based systems [125] enable certain roles to be assigned to the users for data access. ABAC [159] employs cryptographic and non-cryptographic techniques, whereas IBAC uses identity-based encryption mechanisms that utilize user identity for data encryption. Data sharing is a distinctive feature of e-health systems. Data can be shared among various stakeholders such as healthcare providers, hospitals, healthcare organizations etc. Search is an alternate substantial function of an e-health system.

Proxy encryption and public-key encryption are widely used encryption techniques for a data search.

2.4 Cloud Computing Security: State-of-the-art and Research Challenges in e-health

The evolution of the 21^{st} century has witnessed great leaps in digital technology where paper-based records are converted into digitalized electronic records such as electronic medical records (EMRs), electronic health records (EHRs), personal health records (PHRs), and electronic health data (EHD). EHRs and EMRs are the health records of patients handled by healthcare professionals, whereas PHRs carry personal data which is handled and monitored either by the patient or their relatives on a regular basis. EHD as electronic health records or computerised patient records is a systematized collection of the smart health records of patients [156]. These records are comprised of a wide variety of data, such as medical histories, demographics, medication, immunisation status, laboratory test reports and other sensitive patient information. EHD systems have remarkable benefits over conventional paper based records. Unlike paper-based records, EHRs incur less manpower, time and physical storage [75]. The advantages of EHRs include easier and faster clinical data access, the ability to maintain effective clinical workflows, mitigation of medical errors, enhanced patient safety, reduced medical costs and better and stronger support for clinical decision-making. Realising the benefits offered by an EHD system, more than 90% of healthcare institutions in Australia have adopted this system to facilitate effective medical resource allocation and efficient healthcare [75]. The ability of EHD to provide better management of healthcare has been ascertained and testified by various users. However, the transition from conventional healthcare systems to e-healthcare throws unique challenges with respect to the privacy, confidentiality, and security of medical information.

Cloud computing is a recent paradigm in digital technology and is being extensively used in the healthcare industry [59]. It not only provides convenient storage of medical information but also facilitates the easy exchange or transmission of medical data

Table 2.1 Cloud Computing Security Techniques.

Scheme	Advantage	Disadvantage	Reference
Privacy-preserving biometric identification scheme	Maximum data privacy resistant to collusion attacks	Need to trust the cloud service provider, centralised data storage, computationally expensive for real scale problems	[169]
TMACS	Security and system level robustness,ensures security and efficient performance	No attribute revocation function, re-using master key shared among multiple attribute authorities (AA), Computational and communication overhead	[90]
RAAC	Robust and secure access control, resolves single-point performance bottleneck-problem	Need to trust central authority (CA) for key generation and distribution, honest-but-curious cloud servers, AA can be compromised, storage overhead for key generation and auditing, communication overhead on CA and AA	[153]
Identity-based encryption	Reduces encryption complexity	Secure channel required between user and key generator	[30]
Attribute-based encryption	Fine-grained access control, collusion-resistant and minimal communication overhead	Data owner requires each authenticated users' public key to encrypt data	[30]
Attribute based cloud storage with secure provenance	Protects data privacy, fine-grained access control, efficient user revocation, scalability, dynamic user management, data provider anonymity and traceability	Data decryption is expensive due to the complexity in bilinear pairing computations and high data latency	[42]
Audit-free cloud storage via deniable ABE	fine-grained access control mechanism, ensures data privacy	chances of decryption errors, extra overhead of generating deniable keys	[39]
Unified fine-grained access control for PHR in cloud computing	Flexible and fine-grained access control to PHR, reduced encryption decryption costs	Complex key generation, required to trust AA and policy manager, No user revocation, limited to a few users	[89]
PPDP	High level of privacy, highly efficient technique for disease prediction	Computation complexity, communication cost increases with increase in EHRs, verification mechanism is not specified	[161]
Efficient anonymous ABE with access policy hidden for cloud computing	Anonymity, data security, fine-grained access control	Requires a trusted AA, computational complexity and storage overhead due to the addition of fake attributes to the access structure	[62]
Secure data sharing in cloud computing using revocable storage IBE	confidentiality, forward/backward secrecy	system is not scalable, key authority can be compromised	[151]

among various stakeholders. The large- scale proliferation of health information in the age of big data necessitates the burgeoning role of cloud networks not only for hosting unlimited amounts of data but also for its easy access across the Internet [88]. It facilitates the creation, storage and retrieval of healthcare information by all stakeholders viz healthcare providers, doctors and patients with ease, irrespective of the barriers posed by time and space. Cloud services provide immense benefits in terms of cost-effective storage, access, processing and updating of information with improved efficiency and effectiveness. Since the data is running on a wide network of remote servers, which are integrated and operated as a single ecosystem accessed from different locations by multiple users, it is susceptible to intrusion or compromise, thereby posing a threat to privacy and security. Moreover, the majority of medical data is highly sensitive and strictly confidential, so its storage on third-party servers naturally increases these vulnerabilities [1]. Generally, a patient may have several healthcare providers viz primary care physicians, therapists, specialists and several insurer providers for medical, dental, vision etc. [165]. Considering the susceptible nature of health information in the public domain, there is an imminent need to devise a more secure, efficient and effective mechanism for sharing and accessing data among stakeholders.

In the healthcare sector, although EHRs are subjected to various challenges with respect to privacy and unauthorised access, the most prominent pertains to data privacy and security [1]. Risks vary from malware attacks, which compromise the integrity and confidentiality of medical data to distributed denial-of-service (DDoS) attacks which are capable of depriving a system's ability to provide efficient patient care. Cyber-attacks, such as those caused by ransomware, have greater ramifications that go beyond financial loss or privacy breaches [5]. In the USA, hackers broke [55] into the database of community health systems (CHS) of a prominent hospital group and accessed a great deal of personal health information, including the social security numbers of more than a million patients. In a similar incident, Anonymous, an internet vigilante group, targeted several hospitals and launched a DDoS attack on their websites, crippling medical services [2]. These incidents highlighted an imminent need to protect and secure the confidentiality, integrity, availability, security and privacy of protected health information (PHI) as a primary priority in EHR. In this context, the role of cyber

security is paramount in preventing, detecting, and acting on unauthenticated access to health data, and its impact on social, economic, political and cultural conflicts. According to the Health Insurance Portability and Accountability Act (HIPAA), it is the responsibility of healthcare providers to maintain the confidentiality of health data [103]. Several techniques are already in use to secure the security and privacy of smart health systems in the cloud environment.

Some of the advanced privacy-preserving mechanisms that preserve cloud security can be adopted to e-health while some cannot due to security concerns. Cloud computing is a centralized mainframe computing paradigm owned by a cloud provider which is less patient-centric and is prone to insider attacks that makes the health records more vulnerable. This is one of the major downsides of cloud computing. Even though cloud techniques adhere to strict security measures, it does not offer a foolproof solution to be adopted into e-health, taking into account of the security issues. Zhu et al. [169] proposes an efficient privacy preserving biometric identification scheme in which a huge volume of biometric data such as fingerprints, irises, voice patterns and facial patterns are encrypted and outsourced to the cloud to avoid expensive storage and computation costs. The scheme is resistant against collusion attacks and provides a maximum level of data privacy. This approach can be applicable to the e-health cloud for efficient data storage in which health records can be encrypted and stored in the cloud which achieves a certain level of data protection. However, as health records are extremely sensitive and data is exposed to the database owner, this scheme is less acceptable in terms of security. Also, this scheme cannot be considered for EHRs as it is not patient-centric and computationally infeasible for real scale problems. The work in [90] proposes a robust and verifiable hybrid multi-authority ciphertext-policy attribute-based encryption (CP-ABE) access control scheme by combining (t, n) threshold secret sharing and a multi-authority CP-ABE scheme for public cloud storage which improves both security and performance by overcoming the single-point bottleneck problem. Xue et al. [153] propose a robust and efficient access control scheme that resolves the single-point performance bottleneck in most of the existing CP-ABE schemes using an auditing mechanism. Even though these schemes [90] [153] are advanced access control schemes that have high security measures, they cannot be adopted directly to e-health as

these schemes cannot guarantee protection from insider attacks since it is controlled by a central authority and multiple attribute authorities. A special encryption technique named Deniable ABE scheme based on Waters CP-ABE scheme was proposed that allows cloud storage providers to create forged user secrets from stored cipher text to prevent the data from being accessed by outside coercers [39]. This scheme combines the advantage of both ABE and symmetric key encryption as it supports a multi-privileged access control for PHRs by combining the encryption of data from multi-patients that falls under a similar access policy [89]. Zhang et al. [161] propose an efficient privacy preserving disease prediction scheme using a single layer perceptron learning algorithm. This model encrypts the symptom information submitted by the patient and the cloud uses the encrypted prediction models trained by it to diagnose the patient's disease without revealing the patient's privacy. Existing studies explore several encryption techniques to resolve the security issues in cloud computing [97]. Choudhury et al. [41] proposed a strong two-step user authentication process where the user is verified before they enter into the cloud. This technique restricts DoS attacks and provides efficiency to cloud computing. Li et al. [81] proposed identity-based authentication of cloud computing and it services a combination of identity-based hierarchical model and the corresponding encryption and signature. A cryptographic framework for secure data management using ID-based cryptography was proposed by Kaaniche et al. [69] by encoding and exchanging the data with clients so that no malicious user can view it without the owner's consent.

These mechanisms [89] [161] impart a high level of data privacy but they are still impractical for health records due to their computational complexity and scalability issues. Other work presented an anonymous CP-ABE with hidden access policy and provides authorized access control with constant key length [62]. Wei et al. [151] proposed a revocable storage identity based encryption (IBE) that provides forward and backward security of ciphertext. Most of the existing cloud storage systems with secure provenance lack poor access control, incur excessive performance overhead and do not support dynamic user management. This work solves the problem by presenting an attribute-based cloud storage system with secure provenance [42]. Even though ABE schemes are the most efficient of the encryption techniques and provide fine-grained,

well-formed access to health records, they are still impractical for the proper execution on EHRs due to their expensive computation [62] [42], key management complexity and challenges in managing access control policies [39] when the attributes in the access structure grow. Despite the attractive features offered by the cloud, the transition of the healthcare field to the cloud environment increases concerns about privacy, security, access control and compliance due to the inherent security challenges related to cloud technology. Patients lose their physical control by storing health information in cloud servers which can be seen as a threat to patient privacy. Data security and data integrity are also challenging issues when storing and accessing data in the cloud arena [121]. Another downside is that cloud service providers play a vital role in transaction analysis, access control, data protection and service integration. With the advancement of technology, the emergence of advanced cyber threats has escalated, which hinders the privacy and security of EHRs [76]. Therefore, it is very important to guarantee integrity, confidentiality, reliability as well as authenticity of the e-health data in either a private, public or hybrid cloud environment. Consequently, this research introduces the concept of a permissioned patient-centric blockchain for EHRs that eliminates most of the existing bottlenecks in the cloud.

2.5 Classification of Privacy Preserving Mechanisms in Electronic Health Records

This section discusses several research studies that have been carried out on two methods, namely the cryptographic and non-cryptographic approaches, and also discusses their challenges in e-health. Furthermore, several techniques that preserve data security, data privacy and data anonymity in the cloud are analysed. In addition to this, some searchable encryption (SE) techniques are presented to query the encrypted data in the cloud. Since the data is encrypted and stored in third-party cloud servers, normal searching schemes cannot be applied. Searching encrypted data is arduous, so searchable symmetric encryption (SSE) has been proposed that enables keyword searches across encrypted cloud data. Different from recent surveys, our research study

systematically covers all aspects and methods of EHR privacy and security in the cloud. Moreover, the survey also reveals the advanced cloud computing security techniques and their research challenges and at the same time incorporates the potential benefits of the blockchain technique to offset those shortcomings. In addition, we conclude the discussion with the open research problems and future directions which will expand the scope of further research in data security and privacy. The cryptographic schemes employ encryption techniques, namely: symmetric key encryption, public key encryption and several other cryptographic primitives, whereas non-cryptographic approaches include access control mechanisms such as RBAC, ABAC, IBAC etc. The taxonomy of the privacy preserving mechanisms is described in this chapter and illustrated in Fig. 2.2.

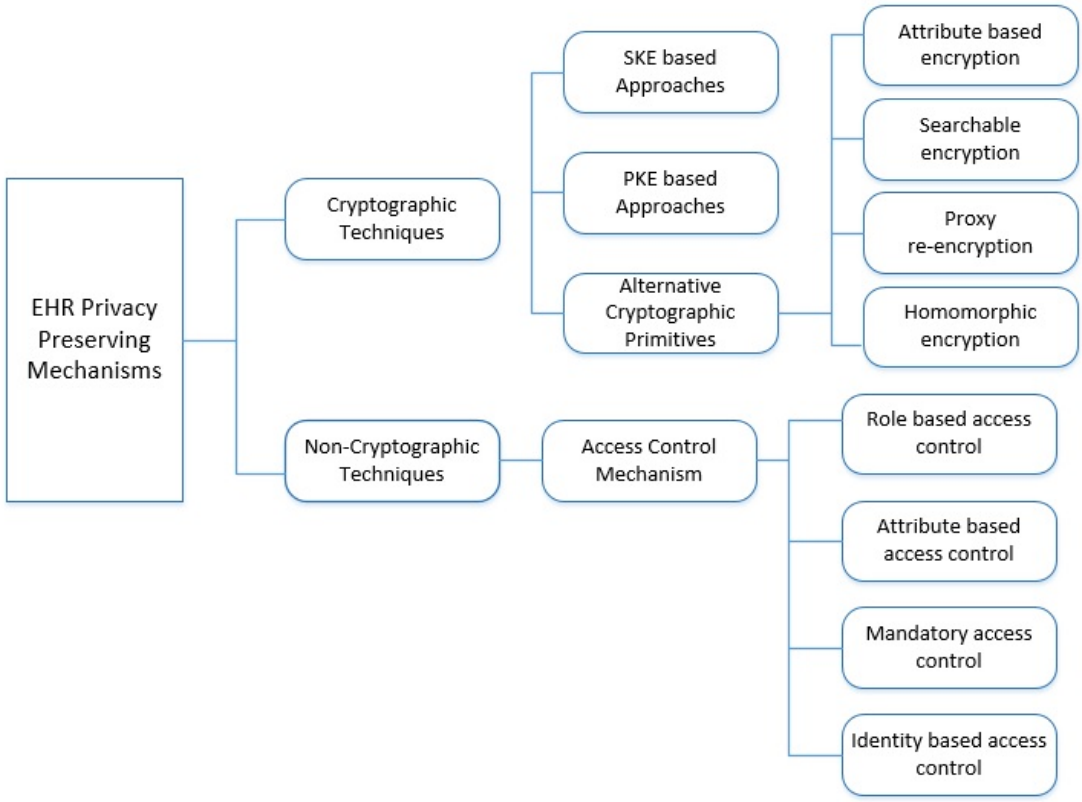


Fig. 2.2 Classification of Privacy Preserving Mechanisms in Electronic Health Records.

2.5.1 Cryptographic Approaches

Cryptographic approaches can be symmetric key cryptography as well as asymmetric key cryptography in which the prior uses the same key for the encryption and decryption whilst the latter uses different keys. This study includes encryption schemes such as symmetric key encryption (SKE) and public key encryption (PKE) and a few alternative cryptographic primitives. In PKE schemes, two different sets of keys are employed i.e. a public key and a private key pair for data encryption and decryption whereas SKE-based approaches utilize a single shared secret key for the same. Alternative cryptographic primitives include several encryption schemes viz attribute based encryption(ABE), searchable encryption (SE), proxy re-encryption, homomorphic encryption, identity based encryption (IBE) etc. Non-cryptographic approaches are associated with a policy-based authorization infrastructure labeled as access control mechanisms viz RBAC, ABAC, mandatory access control (MAC), IBAC etc. This section gives a detailed survey of the significant research works based on SKE, PKE and alternative cryptographic primitives that enforce the security and privacy of electronic health solutions.

2.5.1.1 SKE-based Approaches

SKE employs the same shared secret key for encryption and decryption and it is highly effective in EHR systems. But it introduces the inevitable additional complexity since it requires additional access control mechanisms for the effective sharing of EHR. The commonly used SKE-based algorithms are advanced encryption standard (AES), data encryption standard (DES), stream ciphers such as RC4, A5/1, and Blow Fish etc. Some of the SKE-based approaches are described in the following and a comparison is shown in Table 2.2.

Lee [79] proposed a cryptographic key management protocol based on symmetric cryptosystems to meet HIPAA regulations. The three entities used are the government healthcare office (SG), the server of a healthcare provider (SH), and patients. The main three phases of the scheme include registration, encryption and decryption. Initially, the

Table 2.2 Symmetric Key Encryption Based Approaches.

Sl No.	**Technique(s) used**	**Strength**	**Weakness**	**Privacy Requirements**							**Reference**
				IN	CO	AU	NR	AC	AN	UN	
1	Symmetric Key Encryption (SKE), Smart Card (SC) , Digital Signature	-	Usage of smart card for every access	✓	✓	✗	✓	-	-	-	[79]
2	Symmetric Key Encryption (SKE)	unlinkability among electronic health records	Infeasible Portability technique	✓	✓	✓	✗	✗	-	-	[91]
3	SKE, SC,license file	Data ownership is ensured	Smart card is required for retrieval	✓	✓	✓	✓	-	-	-	[33]
4	SKE	Ensures patients' data ownership	Emergency access provision is insecure	✓	✓	✗	✓	✓	✗	✗	[32]
5	SKE	Key distribution issue is resolved	Difficult to manage multiple user roles	✓	✓	✓	✓	-	-	-	[166]
6	Searchable Symmetric Encryption(SSE), AES	Dynamic searchable symmetric key encryption and resolves key sharing problem	Cannot support multi-keyword search	✓	✓	✓	-	-	✓	-	[82]

* IN: Integrity, CO: Confidentiality, AU: Authentication, NR: Non Repudiation, AC: Accountability, AN: Anonymity, UN: Unlinkability.

patient needs to register with SG to receive a healthcare card which makes him eligible for the medical services offered by SH. The encryption phase involves encrypting PHI by enabling the health data card by entering the user PIN or by biometric verification. This can be done by generating a session key and cryptographic checksum by concatenating the hash value of patients' master key and the session key of healthcare provider. The decryption conducted is two-fold, one with the patient's consent and the other with emergency cases. This can be done by computing the master key and session key of the healthcare provider. A secure EMR sharing scheme was proposed by Li et al. [91] to improve the unlinkability between the patient and the EMR. EMRs are encrypted using symmetric key encryption using a one-time key and records are stored anonymously. Doctors use digital signatures using a private key to process electronic medical records. This approach requires an EMR number i.e. the PID, SID, the identity seed which is stored in the patients' medical card and the random value R, which is created by the doctor to access the EMR of the patient. Each key used in this process is for encrypting one EMR, increasing the confidentiality of each electronic medical record. Since the identity seed SID is based on the smart data card, medical records cannot be read without authorization.

An EHR sharing and integration system was proposed by Chen et al. [33] to protect the EHRs in normal and emergency situations in hybrid healthcare clouds. This approach encrypts each medical record using an individual symmetric key ck using a symmetric encryption scheme in public and private cloud environments. Here, the doctor creates the patients' health record and it is encrypted by the symmetric key ck along with a license L. This license provides an emergency key to access the encrypted data by the cloud even if the server is not provided with direct access. The patient has to give the smart card to the doctor to decrypt their EHR. This design encrypts all the medical records and decryption is possible only by using the patients' private keys in which the private key is split into two parts, whereas one among the keys will be escrowed by the hospital server and the other key will be stored on the patient's smart card. The downside of this approach is that the license file also needs to be encrypted with the hospital's public key. A new dynamic access control scheme for PHR was proposed by Chen et al. [32] under the cloud computing environment. This scheme uses

a Lagrange interpolation polynomial to establish secure PHR information access that ensures security which is suitably scaled for a large number of users. The approach adopted cryptography based on Lagrange multipliers for encrypting the health records ensuring that every patient has maximum control over their medical records. By allowing every patient to generate his/her own related keys, users can choose with whom to share their health records. This reduces key management complexity and at the same time, allows users to not only retain access control of PHR, but also permits issuance of limited access rights to other users, such as doctors, pharmacists, nurses, researchers etc. This approach carries computational overhead. To reduce the complexity of key distribution, this method overhauls past hierarchical models and creates partial order relation to manage users. This is a very flexible approach for multi-user dynamic access control in coordinating the need for the immediate addition or removal of user access and also for the addition and modification of PHR, making it more suitable for PHR cloud application. Zhang et al. [166] present a role-based and time-bound access control (RBTBAC) model which is an integration of RBAC and a time-based access control model that ensures the security and privacy of EHRs on untrusted cloud servers. This model is a logorithmic composition of RBAC and time-bound hierarchical key management in which an authorized user of the EHR system who is allotted a time period can access the data on the basis of his role. This model extends greater flexibility in spatial and temporal capabilities to restrict access to sensitive data. The EHR are encrypted through SKE. This work develops a role-based privacy-aware access control and management of EHR data and also utilizes a time tree method which offers time bound access control and authorization. In this approach, a user is required to work in several roles and also owns and administers multiple keys. It is a requisite to encrypt sensitive medical healthcare records prior to uploading them to the semi-trusted cloud servers. As searching encrypted data is arduous, searchable symmetric encryption(SSE) [82] has been proposed that enables keyword searches across encrypted cloud data. This approach presents a highly efficient and secure dynamic searchable symmetric encryption(SEDSSE) method in medical cloud data by leveraging the secure k-nearest neighbor (kNN) and ABE techniques. This approach used an AES symmetric encryption algorithm to encrypt the documents and shares the symmetric

secret key only with authorized doctors who satisfy the access policy related to ABE.

From Table 2.2, it is evident that even though most of the SKE-based approaches satisfy IN and CO, they still lack AN, UN, AC and NR due for the following reasons. In SKE approaches, both the sender and the receiver are required to trust each other as they will be sharing the same secret key for encryption and decryption that makes anonymity almost impossible. In SKE techniques, non-repudiation and unlinkability would be violated if the user-credentials such as passwords or smart cards were lost, shared or stolen. Moreover, the use of shared user IDs and passwords destroys accountability. Most of the methods in SKE fail to mention the procedure to restore anonymity or the key. Moreover, these schemes are unable to operate in a dynamically changing cloud environment because of its inflexible access control and inability to manage multiple user roles.

2.5.1.2 PKE-based Approaches

PKE approaches entail two separate keys, one public key and one private key. Some of the PKE-based approaches are described in the following and a comparison is shown in Table 2.3.

Autonomous PKE schemes are computationally inefficient because of their slower operation and large key sizes. Therefore, PKE schemes can be more efficient in combination with SKE schemes in which SKE schemes can be used for encrypting the contents and public private key pairs can be used to secure the symmetric keys. This framework [64] uses a public key infrastructure (PKI) to address diverse security requirements such as authentication, confidentiality, integrity, access-control, non-repudiation etc whereas the EHR are encrypted using a shared symmetric key generated by healthcare providers. PKI binds public keys with unique user identities which consist of digital certificates, a registration authority, a certificate authority, a certificate repository database and a certificate management system. This proposed architecture builds a secure EHR sharing framework that ensures effective sharing of EHRs between patients and several healthcare providers. Authentication between the

Table 2.3 Public Key Encryption-Based Approaches.

Sl No.	Technique(s)	Strength	Weakness	Privacy Requirements							Ref.
				IN	CO	AU	NR	AC	AN	UN	
1	PKI,SKE,Digital Signature	Secure Electronic Health Record Sharing	Incompatibility in different EHR representations	✓	✓	✓	✓	-	-	-	[64]
2	Public Key Encryption (PKE), Digital Signatures	Secure EHR referral	Inadequate patient centric functions	✓	✗	✓	✓	✓	-	-	[68]
3	PKE, Signature verification	Patient control over health data	Misuse by record issuer	✓	✓	✓	✓	✓	-	-	[100]
4	ElGamal PKE, PKI	Resilient against inside attacks	Expensive computation, Not suitable for dynamic access control policy	✓	✓	✓	✓	-	✓	-	[156]
5	Broadcast ABE, PKE with keyword search	Efficacious user revokement	Obstinate access control	✗	✓	✗	✗	✗	-	-	[110]
6	PKE, Digital signature	Trusted Virtual Domain utilization	Scalability issues	✓	✗	✓	✓	-	-	-	[135]
7	PKE, pseudonymity	Anonymity between user and provider	Service provider may misuse health data contents	✓	✓	✓	✓	✗	✓	✓	[95]
8	Homomorphic encryption, Probablistic algorithm	Security of medical images	Restricted to medical image processing	✓	✓	✓	✓	✗	-	-	[111]
9	PKE with keyword search	Address key management problem, Key escrow problem, min computational cost and complexity	Requires a trusted key generation center, Insider attack is possible	✓	✓	✓	✓	✗	-	-	[136]

EHR sharing cloud and healthcare providers is achieved by signing the documents with the sender's private key so that only the targeted healthcare provider can verify the signature to retrieve the equivalent health records. PHR privacy is ensured in this framework [68] by creating a security model called the Online Referral and Appointment Planner (ORAP) in which medical information is encrypted at the client side. In the ORAP model, EHRs are cached in a trusted environment, i.e. at the physicians' practice locale. EHRs are encrypted by the public key of the receiving entity and signed before being transmitted to the cloud and decryption is restricted to authenticated entities only. This framework used the Amazon S3 cloud for temporary storage and German healthcare telematics infrastructure components to provide secure and strong encryption and signatures for all documents transferred to the patients' health record. Several PKE-based approaches are compared in Table 4.

Mashima and Ahamad [100] designed a patient-centered monitoring system to safeguard the risk of storing and accessing electronic health information in the cloud. This work developed a system that allows patients to have explicit or implicit control regarding when and how their medical information is accessed. Health records are encrypted through PKE with the associated hash values [135]. Universal Designated Verifier Signatures (UDVS) which generate a designated verifier signature is also introduced as part of this work to ensure patient record usage is restricted to authorized entities. The main drawback with this system is that the confidentiality of the record is compromised as the health data is initially built by an issuer who has information about the details of record, hash values, and signatures. One of the prominent works mentioned in the literature is that of Xun Yi et al. [156] which provides a multi-party framework to ensure patient privacy in which all the EHRs are encrypted with a common public key and decryption needs the cooperation of all concerned parties. This approach is constructed on PKI based on the ElGamal Threshold public key encryption scheme [156]. This scheme uses modular exponentiation which is less computationally expensive and re-encryption is not required. This prevents any server and collusion of up to $n - 1$ servers and therefore is robust against internal and external attacks and also achieves n server joint authentication over only one database. Narayan proposed a cloud-based EHR system by integrating [110] symmetric key cryptography, public key

cryptography and attribute-based encryption. In this approach, medical data is encrypted by a patient's symmetric key and the metadata file which describes information regarding access policy. Location information is encrypted using broadcast CP-ABE before being stored in the cloud. This approach supports direct revocation without data re-encryption but incurs additional costs on the patient side since re-encryption and updating access policies are borne by them. Another drawback is that all the encrypted files can be accessed by the trusted authority.

A solution to address the security issues is to use a security architecture on Trusted Virtual Domains(TVDs) in the e-health infrastructure. The work in [95] made use of TVD to establish access control by employing three privacy domains: trusted, e-health and untrusted domains. TVDs are a collection of different virtual machines that have common security policies and trust each other. TVD systems have the advantage of flexibility when integrated with legacy systems. This approach makes use of PKE encryption for storing and transmitting e-health data in external storage. The main drawback associated with this approach is the complexity in deploying the TVD-based solutions and the scalability issues where these domains are executed on a host computer. Pecarina et al. [111] described a PKE-based framework to enhance privacy by providing anonymity in data storage and efficient access control to authorized collaborators in a semi-trusted health cloud [136]. PHRs will be encrypted by the patient using the public key of a cloud service provider (CSP) prior to storage in the cloud. The patient records are decrypted by the CSP using its private key. After storing the PHRs at a location, the location is finally encrypted through the SKE of the CSP. The work in [143] proposed an efficient homomorphic encryption for the encryption of medical data images without hindering data confidentiality. A probabilistic algorithm is used for both key generation and encryption. This approach stores images in a standard format, namely Digital Imaging and Communications in Medicine (DICOM) and converts the input image into a matrix followed by performing key generation based on the homomorphic property and encryption using homomorphic public key encryption before transmitting to the cloud. Efficiency of the data is performed using peak signal to noise ratio (PSNR) and mean square error (MSE) analysis, histogram analysis, and correlation analysis etc. An efficient key word search mechanism which employs a

public key encryption was proposed by Ma et al. [96] for a flexible healthcare system in cloud servers. This approach constructs an encrypted keyword index with users public key attached to encrypted health data prior to uploading to the cloud server. It makes use of a trusted key generation center to generate the master key, public parameters and the user's partial private key. This work addressed key management problems and key escrow problems with minimum computational cost and complexity.

From the discussion, indisputably PKE schemes in the cloud are computationally inefficient to some scenarios due to their larger key sizes. Some of the existing PKE techniques fail on the confidentiality of health data as it is compromised by an authorized entity who exploits data ownership. In some PKE techniques, authenticity is not satisfied considering all the encrypted files are accessible by the trusted authority who exploits trust. Many of the public key systems use a third party called a certification authority (CA) to digitally sign their public key, turning into a digital certificate to make it safe. However, if the CA is compromised, masqueraders can attack so the data will be sent to a wrong destination. Furthermore, public key cryptography can encrypt data only up to the key size, hence the distribution of public keys is troublesome in environments which handle large data sets. While some schemes are designed to protect against insider attacks, other schemes focus on patient-centered PHR in which the records are first created by record issuers who know the content of the records, corresponding hash values and signatures. Consequently, inside attacks can occur when an issuer himself misuses the health records created by him, forfeiting data integrity. Compromising secret keys Sk of the patient and monitoring by a third-party results in a loss of data confidentiality. In addition, some other schemes also discuss that the PKE technique has a slightly higher computational cost due to the re-encryption of records when updating access policies.

2.5.1.3 Attribute-based Encryption (ABE) Approaches

This section overviews the alternative cryptographic approaches to securing privacy in e-health clouds. The primitives include ABE, SE, IBE, homomorphic encryption, proxy

Table 2.4 ABE-based Approaches.

Sl No.	**Technique(s) used**	**Strength**	**Weakness**	**Privacy Requirements**							**Ref.**
				IN	CO	AU	NR	AC	AN	UN	
1	PKE, ABE	Flexible Access control	Linkability among Electronic Health records	✗	✓	✗	✓	✓	-	-	[110]
2	Broadcast ABE, PKE with keyword search	Effective user revokement	Rigid access control	✗	✓	✗	✓	✗	-	-	[20]
3	KP-ABE,PRE,Lazy re-encryption	Scalable access control	Computational overhead	✗	✓	-	✗	✓	-	-	[158]
4	CP-ABE, IBE, digital signatures	patient-centrical access policies	Communication delays	✓	✓	✓	✗	✗	✗	✗	[63]
5	ABE	Maintains user anonymity of data storing entities	Cloud is aware about health record access policy	✗	✗	✓	✗	✓	✓	-	[23]
6	MA-ABE	Efficient user revocation	Restricted access policy specification	✗	✓	✓	✗	✓	-	-	[96]
7	Hierarchical-ABE with Keyword Search, Proxy re-encryption	Fine grained access control, versatile client revocation	Single keyword search is possible	✓	✓	✓	✗	✓	-	-	[16]

re-encryption etc. Attribute based encryption introduced by Sahai and Waters is based on public key encryption to protect cloud data where the encryption and decryption is done on the basis of user attributes [122]. In ABE, encryption is based on the access-structure policy in which the cipher text can be decrypted only when the user attributes match the ciphertext attributes. The two main types of ABE are ciphertext-policy attribute-based encryption (CP-ABE) and key policy attribute-based encryption (KP-ABE). In KP-ABE, the access policy is enciphered in the user's secret key and the decryption of the ciphertext is possible only when the user attribute matches the access policy, whereas in CP-ABE, the private key of each user is tied to a set of attributes and a ciphertext is associated with a universal set of attributes which can be decrypted when the user attributes match the access policy [122] [20].

This ABE-based approach [63] preserves the confidentiality of EHR by using PKE for scalable authorization. The smartcard of the patient generates a transaction code (TAC) which is the authorization secret, before the medical data is uploaded to the cloud server. PKE is used for authentication and the patient's smart card and TAC are used as authorization. The health professional needs to enter the TAC to encrypt the medical data and the encryption/decryption function generates a public key for encryption which is the hash value of the patient's identity and TAC. The decryption can be performed using TAC and authentication from a private key generator (PKG). The problems in achieving confidentiality, scalability, and fine-grained access of outsourced data in the cloud are enumerated by Yu et al. [158]. This approach resolves problems, including key distribution and data management issues, by combining techniques such as ABE, KP-ABE, proxy re-encryption (PRE), and lazy re-encryption as a hybrid encryption scheme to secure fine-grained access control. The data encrypted by a single user will be shared among different users by key distribution. In this approach, re-encryption of the data files and updates of secret keys are consigned to cloud servers. A copy of the user's secret key is kept with the cloud servers to update the secret key components and re-encrypt the data files. Lazy re-encryption is used to reduce computational overhead in cloud servers. It can prevent the revoked users from capturing the updated information once the file contents and keys are modified post-user revocation. A patient-centered cloud-based EHR system that integrates symmetric key cryptography, public key

cryptography and an attribute -based broadcast ciphertext policy attribute-based encryption (bABE) architecture is proposed in [110]. This method allows for the encryption of health data using a symmetric key and metadata files that include a description of the file and an attribute-based access policy. Location-based information is encrypted using broadcast CP-ABE by the patient and enables them to store this within a cloud platform. This approach also includes a key word search functionality by amalgamating bABE and PKE with keyword search (PEKS) [23] to carry out private searches in encrypted data without unveiling the matches to the cloud. Even though this approach facilitates direct revocation without data re-encryption, it incurs additional computational costs as the re-encryption and the updating of access policies are borne by the patient. An additional drawback exists in the internal vulnerability of access to encrypted files by the trusted authority without reference to a permissioned user. Some of the ABE-based approaches are described and a comparison is shown in Table 2.4.

The efficient and secure patient-centric access control scheme (ESPAC) [16] for the cloud using CP-ABE ensures PHI privacy, permitting data requesters to access the health data in accordance with role-based access privileges. For secure communication between a remote patient and the e-health cloud provider, IBE is employed, where the access control is handled by CP-ABE. Ruj [120] presented a novel technique using an ABE-based access control mechanism that maintains user anonymity for storing PHRs in the cloud. The user identity is unknown to the cloud but the verification of the user credentials and communications between users and the cloud are secured by secure shell protocol (SSH). This approach is collision resistant and is resistant to replay attacks and has a decentralized key distribution.To facilitate flexible and effective access control for PHR, this scheme suggests an efficient patient centric framework [87] which employs ABE to encrypt a patient's PHR file before uploading to the cloud.This scheme provides several data owner settings and also categorizes the PHRs into two different sub-domains viz public and private to address the key management hurdles. This approach [58] instigates hierarchical attribute-based encryption with a keyword search scheme that ensures the confidentiality of EHRs in the cloud environment. This scheme encrypts a single access structure in which the trusted authority will issue public and private key pairs. The access policy and time period is set by the information owner

before outsourcing the data to the cloud. A proxy re-encryption scheme is also implemented to deny access after the predefined time period defined by the information owner. This work ensures fine-grained access control, versatile client revocation and lower storage and encryption time costs compared to other systems.

Even though ABE provides dynamic access control and key management, it still experiences some drawbacks. One of the limitations of ABE is that the data owner needs to use the authenticated users' public key for encryption [30]. The drawback with KP-ABE is that the owner of the data cannot decide who can decrypt the encrypted data as the data owner has to trust the key issuer and also suffers poor scalability issues. Consequently, the non-repudiation cannot be guaranteed. In CP-ABE, attribute management and key distribution are managed by a trusted authority. ABE schemes are the most efficient of the encryption techniques and provide fine-grained and well-formed access to health records but they are still infeasible for proper execution on EHRs due to their expensive computation key management complexity and challenges in managing access control policies when the attributes in the access structure grow [36]. Another downside is that most of the ABE schemes use a semi-trusted entity which manages the servers and provides cloud services for this reason, they become a threat to data integrity.

2.5.1.4 Searchable Encryption(SE)

Due to the massive growth of big data, there is large-scale outsourcing of data into cloud servers. As medical data and EHRs are outsourced to remote cloud servers that are exposed to cloud service providers, this leads to various attacks such as DoS attacks or adversary attacks that destroys data confidentiality in the cloud. To protect data and prevent information leakage, cloud data need to be encrypted. Since health data is encrypted and stored in third-party cloud servers, normal searching schemes cannot be applied. It requires some searchable encryption implementation to query the data, as shown in Fig. 2.3. As searching encrypted data is arduous, SSE has been proposed to enable keyword searches across encrypted cloud data. This poses challenges such as:

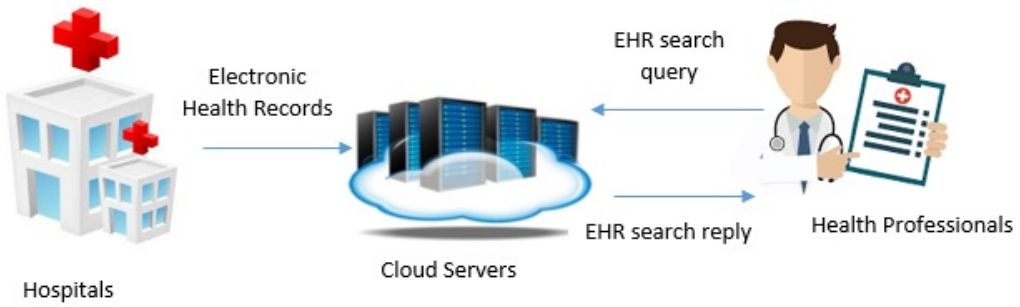


Fig. 2.3 Searchable Encryption.

(1) How does the data owner give search permissions to the data user? and (2) How do the authenticated data users search the encrypted stored data? One of the solutions to these questions is SE. SE is a cryptographic primitive that permits search operations over encrypted data without disclosing information to untrusted servers. These search operations are performed on encrypted ciphertext with the support of a trapdoor function from the user. The main two types are symmetric searchable encryption and asymmetric searchable encryption [112]. Here we discuss SE and categorize its use cases into four

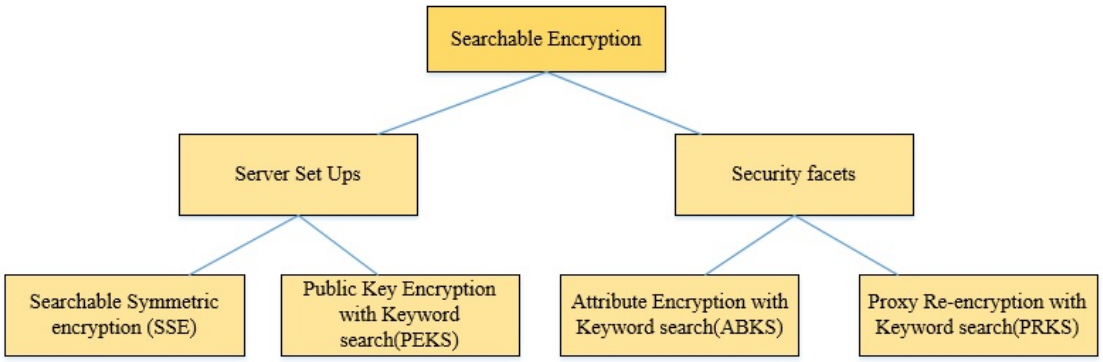


Fig. 2.4 Searchable Encryption.

schemes viz searchable symmetric encryption (SSE), public key encryption with keyword search (PEKS), attribute-based encryption with keyword search (ABKS), proxy re-encryption with keyword search (PRKS) as shown in Fig. 2.4 and their comparison is presented in Table 2.5 and Table 2.6. A searchable encryption service contains three types of entities: a data owner, a data user (data users), and the untrusted cloud [167]. The data owner is a cloud service user who has outsourced the original data to a third-party cloud. Different healthcare application scenarios require different searchable encryption schemes. We can divide existing healthcare application scenarios

into four categories: (1) When the outsourced data are searched only by the data owner, where the data owner is the only authorized data user to search the encrypted data, SSE schemes can be applied in this scenario. (2) When the outsourced data are shared with another user, i.e. there is only one authorized data user who can create the search tokens and search the encrypted data, PEKS schemes are suitable for this one-to- one scenario. (3) When the outsourced data are shared with several users, i.e. more than one authorized user has the permission to search the encrypted data, ABKS schemes can be used in this one-to-many scenario. (4) When the data owner is unavailable and cannot grant search authorization in an emergency, it needs an authorized delegated user to re-authorize the search permission to other user(s) on behalf of the data owner. PRES schemes are applicable to this authorization-delegation scenario [167].

Searchable Symmetric Encryption (SSE)

SSE is a symmetric key encryption technique which outsources data confidentially from one party to another by providing selective search capabilities. This model uses proxy re-encryption that shares medical data in the cloud with end-to-end data encryption that limits data access to authenticated recipients only.This approach preserves the privacy and security in e-health systems with a new cryptographic technique named the conjunctive keyword search with designated tester and timing dependent SE schemes named proxy re-encryption function (Re-dtPECK) [36]. The EHR documents are encoded by symmetric encryption algorithms and a symmetric key is encapsulated with the patient's public key by key encapsulation. This makes use of a delegation function θ to perform operations and uses a conjunctive keyword search mechanism. This approach proposes a novel SSE scheme [142] which enables searching according to the unique keywords stored on the server. The search time is logarithmic and the client can search and update the document whenever required. This makes use of two variant schemes in which the first one is an interactive scheme and the second is non-interactive in which the former needs two rounds of communication for the index generation, updates, and search whereas the latter can be deployed using a hash chain. This method [28] uses an SSE procedure which supports conjunctive search and Boolean queries on stored data which is symmetrically encrypted and focuses on a single keyword search mechanism. This model provides higher security and scales to very large databases. By preserving

keyword privacy, this approach [84] validates and resolves the issue regarding fuzzy keyword searches across encrypted data in the cloud. Fuzzy keyword searches enrich system utility by providing matching files or the nearest possible matching files for user input with the predefined keywords based on keyword similarity semantics, otherwise. This solution precomputes fuzzy keyword sets with edit distance to evaluate keyword similarity and also minimizes the storage and representation overheads by developing an advanced mechanism on constructing fuzzy keyword sets.

Public Key Encryption with Keyword Search (PEKS)

PEKS is a cryptographic approach that uses a public key system to search across encrypted data. Boneh et al. [24] proposed PEKS as an initial scheme which does not uncover any information pertaining to a user's search in the public-key setting and with lower communication complexity. This approach [12] addresses three main issues of a PEKS scheme viz removal of a secure channel, refreshing keywords, and processing multiple keywords. The idea of PKE with a registered keyword search (PERKS) was presented by Tang and Chen [138]. This scheme provides flexibility in such a way that the sender is able to register a keyword with the receiver prior to the sender generating a tag to build searchable content. This makes the scheme more efficient and secure against offline keyword-guessing attacks.

Attribute Encryption with Keyword Search (ABKS)

ABKS is a cryptographic searching approach which uses attribute-based encryption for data encryption. This searching technique permits keyword searches over encoded EHR data by authorized users whose attributes fulfill the access policy. Yang [154] proposed a multi-sender and user scenario that enhances fine-grained access control and supports flexible user revocation using a flexible keyword searching technique and attribute-based encryption. This scheme introduced a novel fundamental named the attribute based searchable encryption with synonym keyword search function (SK-ABSE). An ABE scheme described by Li et al. [83] implements keyword search functions with outsourcing key-issuing and outsourcing decryption (KSFOABE). In this scheme, the cloud service provider undertakes partial decryption tasks assigned by the data user without having any information regarding the plaintext which is secure and robust against the chosen plaintext attack. The verifiable attribute-based keyword search

Table 2.5 Comparison of SE techniques (SSE and PEKS)based on Server Set Ups.

Scheme	**Set Ups**	**Main Operations**	**Query Type**	**Performance**
Re-dtPECK scheme [155]	Multiple server	Time dependent search,symmetric encryption,proxy re-encryption	Conjunctive keyword	Higher security and confidentiality, overcomes keyword guessing attack (KGA)
SSE for Boolean queries [28]	cloud server	Symmetric sncryption, Diffie Hellman	single keyword	higher security, scales to very large databases, moderate data leakage
Fuzzy keyword search [84]	cloud server	Symmetric encryption, edit distance	fuzzy keyword search	Privacy preserving system
PKE with keyword search [23]	single server	Public key encryption, homomorphic encryption	Multiple keywords	preserves privacy, less communication complexity
Trapdoor privacy in PKE [9]	cloud server	Asymmetric searchable encryption, PEKS, IBE	multiple keywords	enhanced trapdoor privacy, key unlinkability
PKE with registered keyword search [138]	single server	PEKS, bilinear pairing	single keyword	Secure against offline KGA attacks, less computational complexity

(VABKS) [168] solution permits a data user to only search over the data owner's outsourced encrypted data whose credentials match with the data owner's access control policy . Liu et al. [92] presented a new approach called key policy attribute-based keyword search (KP-ABKS) which removes the secure channel to validate the searched result from the cloud to reduce the computation complexity on VABKS.

Table 2.6 Comparison of ABKS and PRKS Techniques based on Security Facets.

Technique	**Security facets**	**Query Model**	**User revocation**
Yang [154]	Effective data security	synonym	Yes
Li[83]	Secure against chosen plaintext attack	single	No
Zheng [168]	Secure against chosen keyword attack, keyword secrecy	single	No
Liu[92]	Secure against offline guessing attack, keyword secrecy	Single	No
Shao[128]	Keyword privacy, message privacy	Single	No
Fang[53]	Cipher text security	Single	No
Shi[130]	Selective chosen keyword security	Single	No

- **Proxy Re-Encryption with Keyword Search (PRKS)**

PRKS is a cryptographic fundamental that uses a proxy re-encryption system for searching encrypted data. It permits an authenticated data user who permissions the search capability to other users by re-encrypting the outsourced data [167]. The proxy re-encryption with keyword search functions (PRKS) is the union of two schemes, proxy re-Encryption (PRE) and PEKS. This approach [128] provides two security concepts for bidirectional PRES (Proxy Re-encryption Scheme) : privacy for keywords and privacy for messages. In keyword privacy, the opponent is permitted to obtain the plaintext of any ciphertext and almost all trapdoors, excluding those which are connected to the two specific keywords. Nevertheless, it cannot determine which keyword matches a given ciphertext. This security idea ensures that the test can only be done by the person who has the trapdoor or token. For message privacy, the opponent is permitted to obtain the plaintexts of almost all ciphertexts, excluding one and all the trapdoors, but it cannot determine which message matches with a particular plaintext. This security concept ensures that the one who holds the private key can decrypt the ciphertexts. A new cryptographic approach described by Fang et al. [53] called conditional proxy re-encryption with keyword search (C-PRES) is an association of C-PRE and PEKS. This approach offers various benefits over previous schemes, such as

chosen-ciphertext security, non-interactivity keyword-anonymity, unidirectionality, and collusion-resistance. Shi et al. [130] presented an approach in which the encrypted data will be outsourced to the cloud by the data owner to perform the keyword search on encrypted data with the specified search token. The idea is to combine ABE and PRE in which the data owner permits keyword searches over encrypted data to authenticated users in accordance with access control policies.

We have provided a survey of searchable encryption techniques for healthcare applications. However, none of the existing multi-user SE schemes are practical with respect to the performance required by critical real-world applications and do not scale well for extensive databases. We categorize and compare the different SE schemes in terms of their security, efficiency, and functionality. However, SSE is not the preferred method for querying the search in EHR due to key management issues. Nevertheless, PEKS and PRKS exhibit better performance in terms of security and privacy and are commonly adopted to EHR that supports the search functionality [36].

2.5.1.5 Proxy Re-Encryption

Proxy re-encryption is a cryptographic approach that permits a semi-trusted proxy server to re-encrypt the ciphertext which is encrypted by one user's public key into another ciphertext i.e. encrypted by the public key of another user. For example, Alice sends a message (M) to Bob through a semi-trusted proxy server without sharing Alice's private key to either the proxy or Bob, and without disclosing the secret message to the proxy shown in Fig 2.5. Yang introduced a novel cryptographic approach called the conjunctive keyword search with a designated tester and a timing-enabled proxy re-encryption function, Re-dtPECK, which uses a delegation indicator θ to perform operations and uses conjunctive keyword for the searching mechanism. This scheme proposes a proxy re-encryption mechanism for on-the-road emergencies that permits an emergency medical center to decrypt a patient's health records with the aid of cloud servers and user credentials without disclosing the secret key [118]. Timing-enabled

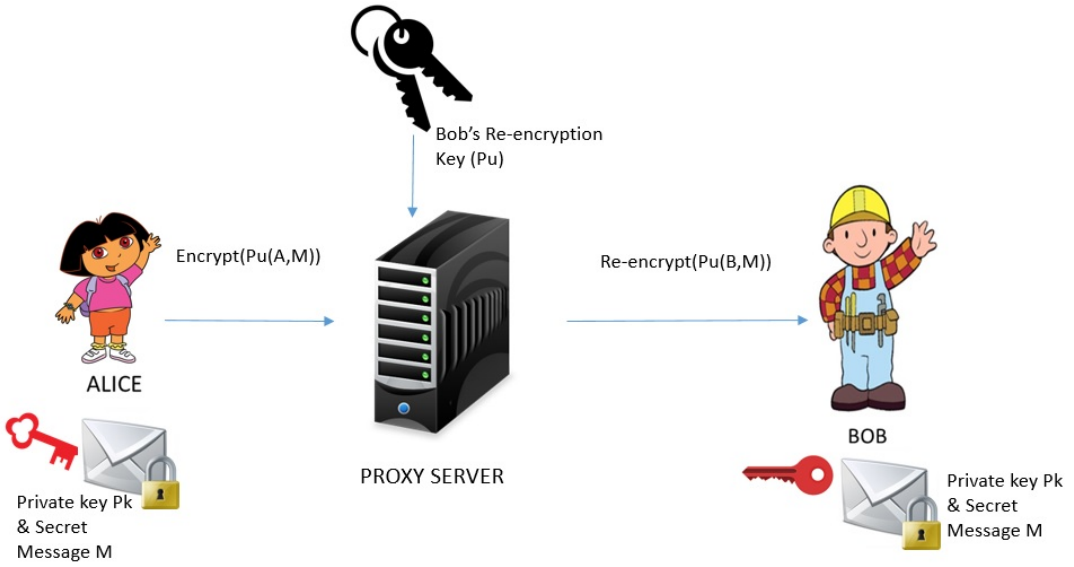


Fig. 2.5 Proxy Re-encryption.

proxy re-encryption systems over conjunctive keyword search have been proposed [21] which allow users to access patient records under a predefined time interval, T. This technique achieves objectives such as efficient access control, user revocation, efficiency, and time-based revocation.

2.5.1.6 Homomorphic Encryption

Homomorphic encryption is a type of encryption which performs computations on ciphertexts in which the data is acquired in an encrypted format and when decrypted, returns the result of operations if they had been performed on the plaintext. A simple example of homomorphic encryption is shown in Fig. 2.6. Barni et al. [15] introduced a multiparty approach for processing an encrypted electrocardiogram (ECG) using homomorphic encryption to preserve patient privacy. Privacy preserving attribute based authentication systems have been introduced for e-health networks which contribute users' verifiable attributes to authenticate users in an e-health system [60]. The proposed scheme relies on homomorphic encryption to guarantee data security, which preserves the privacy of attributes but the computation cost is extremely high. Gentry [57] proposed the idea of fully homomorphic encryption which permits a random number of

additions and multiplications over the encrypted data, whereas, somewhat homomorphic encryption (SwHE) executes restricted numbers of homomorphic operations by evaluating circuits of specified depth. Fully homomorphic encryption based approaches are impractical because of their inefficacy. Lauter et al. [107] presented SwHE to

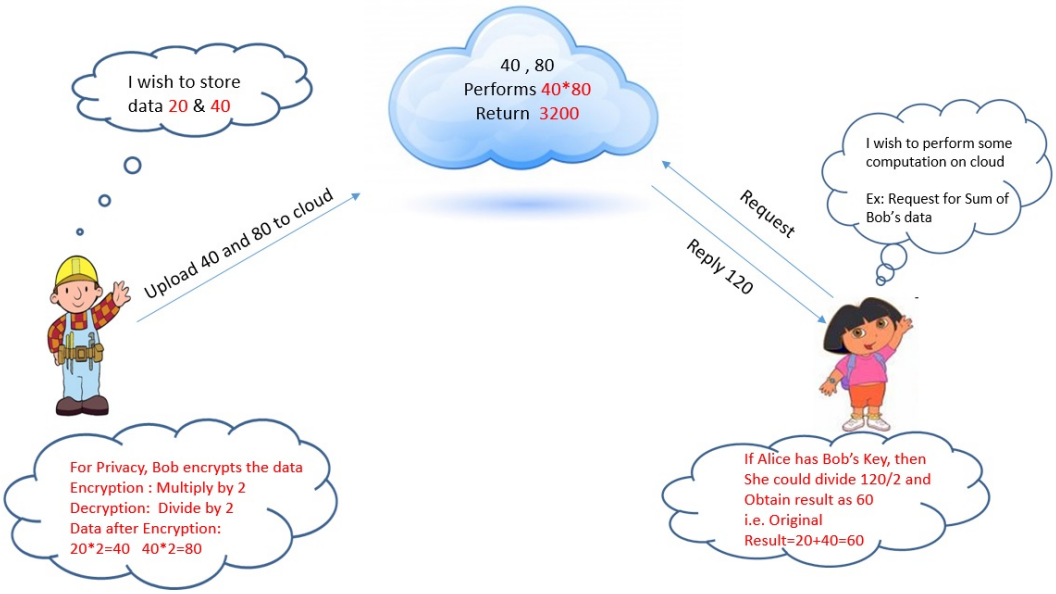


Fig. 2.6 Homomorphic Encryption.

perform computations over the encrypted data. This approach [51] implements a hybrid architecture that uses homomorphic encryption and Rivest-Shamir-Adleman (RSA) to enhance e-health data security on private cloud OpenStack platforms. This architecture enables cloud clients to take control of their cryptographic operations and key management rather than the cloud provider. Sergiu et al. [27] designed a privacy preserving diagnosis model using homomorphic encryption which processes data without allowing any information breach to the cloud provider. Data will be encrypted with the private key of the user before uploading to cloud servers and data evaluation will be done on encrypted data in which the results are oblivious to the cloud. This approach integrates state-of-the art components such as, trans-ciphering, automatic compilation, parallelisation, and message packing, to preserve user privacy.

2.5.2 Non-cryptographic Approaches

Non-cryptographic approaches mainly use policy-based authorization infrastructure such as access control policies to enforce the privacy control over the data. In EHR systems, data access is of a highly confidential nature and data is housed on third-party severs. Access control mechanisms are inevitable and vital as encryption approaches. In a health care information system, access control offers fundamental security barriers to data privacy which limits the access and operation of documents in the EHR system. Some of the main access control techniques are depicted in Fig 2.7. A comparison of a few privacy preserving non-cryptographic mechanisms is shown in Table 2.7. Discretionary access control (DAC) is a form of access control in which the object's

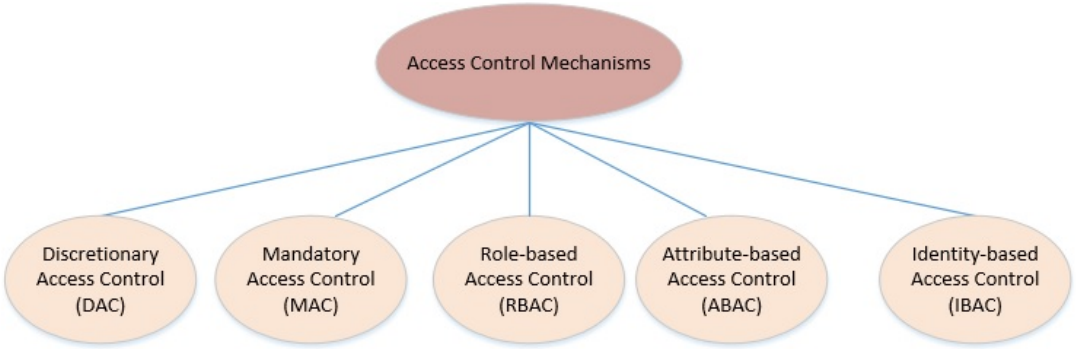


Fig. 2.7 Classification of Access Control Mechanism.

owner has total control over the programs. DAC gives access to objects based on the subject's identity [114]. In MAC, access policy decisions are not made by the individual owners of an object but by a central authority and also the owner cannot change access rights [61]. RBAC defines access decisions on the basis of their job functions in which roles have been allocated to subjects, and the roles are associated with permissions that define which actions can be operated over which objects. RBAC is defined in terms of five basic datasets, namely subjects, roles, objects, operations and permissions. In the context of healthcare, roles could be doctor, nurse, staff etc. and operations can be read, write, add, modify and update records. ABAC is an authentication-based access control in which the decisions for access are made according to the set of user -defined attributes and requesters will be given object access according to the attributes that

satisfy the policy rules. IBAC is an approach to regulate access on the authenticated identity of an individual.

Khan and Ken [72] proposed a context sensitive fine-grained access control mechanism of personal health information by means of discretionary access control and RBAC models. This approach uses eTRON architecture in which authentication is performed using public key cryptography and secure key sharing is established through the Diffie-Hellman algorithm. Harsha et al. [115] presented a patient-centric attribute based method in which each PHR file is encrypted and stored along with an attribute based access policy in an e-health cloud that controls the access to the particular resource and also utilizes a proxy re-encryption technique that helps the authenticated users to decrypt the appropriate PHR files. R.Sandhu et al. [123] proposed RBAC in which the roles have been assigned to subjects and roles are also associated with permissions that define which actions can be operated over which objects. This scheme has several drawbacks. It is an expensive process to define and structure the roles, and it only supports policies that are static and defined in advance. Furthermore, it cannot support dynamically changing environments [77] and also RBAC's coarse granularity causes internal attacks [40]. Yuan and Tong [159] proposed ABAC in which specific attributes of each subject are used to explain access policies for access permission. ABAC resolves issues of RBAC but it has two problems. Initially, ABAC is arduous because of the large number of rules that are required to be examined for access decisions, and secondly for n attributes ABAC may require 2^n rules [159]. Tang et al. [139] introduced a role-based access control (RBAC) model that involves two elements, the client component and the proprietor component in which users are required to correspond with the administration supplier to receive asset access permissions from the proprietors. Sun et al. [133] proposed a usage access control mechanism with purpose extension different from conventional models with respect to its access decision that can be applied in the field of e-healthcare system.

A secure attribute-based access control technique for EHR was presented by Harsha et al. [116] using selective disclosure of the attributes in which the access decisions are made in such a way that the user must acquire the same attribute set that satisfies the defined access policy to the requested resource. This approach employs a public key

infrastructure(PKI) to establish a secure channel to authenticate the health center. This model [131] integrates several mechanisms such as RBAC and ABAC to provide confidentiality for electronic health records. A framework that introduces the concept of a provenance-based access control combined with RBAC with a distributed rule-based mechanism is proposed [78] to enhance the security of cloud data. Bahga et al. [13] proposed an EHR architecture that attains semantic interoperability between stakeholders. This framework adopts a two-level modelling that provides better security and addresses the key requirements of HIPAA and HITECH (Health Information Technology for Economic and Clinical Health act). For secure data storage and secure access a cryptographic model for EHR systems has been proposed [113]. Location awareness and biometric authentication techniques are used for user authentication and steganography techniques are used to conceal EHR data in the cloud by embedding in ECG signals.

Gajanayake et al. [56] presented a new access control technique to preserve patient privacy and confidentiality for EHR by combining three prevalent techniques, namely MAC, DAC, RBAC along with a purpose-based access control. The work in [26] adopted an XACML (Extensible Access Control Markup Language) ABAC mechanism for the protection of EHR against unauthorized intruder access, which supports interoperability. This approach makes use of semantic technologies and an inference engine which uses attributes as classes and rule-based policies for decision making. Seol et al. [127] proposed an EHR model that combines ABAC using XACML to preserve patient privacy and ensure security in the cloud environment. This work makes use of partial encryption based on XML and XML digital signature technology for authentication purposes. An attribute-based access control scheme [117] for an e-health environment, integrated with controlled access delegation, has been proposed. This approach also performs multilevel access delegation with on-demand attribute revocation mechanisms. An authentication algorithm and RBAC to preserve patient privacy in smart health systems [140] has been proposed. It makes use of three parties, namely the health authority, healthcare professionals, and the information consumer. Liu et al [94] introduced an RBAC scheme for EHR on the basis of two roles, one for patients and the other for medical staff. Patients are identified by their identity whereas

Table 2.7 Comparison of Privacy Preserving Non-Cryptographic Mechanisms.

Sl No.	**Technique(s)**	**Strength**	**Weakness**	**Privacy Requirements**							**Ref.**
				IN	CO	AU	NR	AC	AN	UN	
1	RBAC	Simpler access administration	Expensive process to define roles	✗	✗	✓	-	-	-	-	[72]
2	ABAC	Dynamic access control policy	Requires large no: of rules	✗	✓	✓	-	-	-	-	[159]
3	BLAC	Combines advantages of RBAC and ABAC	Security threats	✗	✓	✓	-	✓	-	-	[6]
4	RBAC, AES, SSL, MAC	Semantic interoperability, Scalability	Inflexible access control	✓	✓	✓	✗	✓	-	-	[13]
5	RBAC, PKI	Context and location awareness	Key exchange problem	✓	✗	✓	✗	✓	-	-	[113]
6	MAC,DAC, RBAC,FT PBAC	Combines three access control models	-	✗	✗	✓	✗	✓	-	-	[56]
7	ABAC (XACML)	Flexible access control	Lack of Confidentiality and Integrity	✗	✗	✓	✗	✓	-	-	[26]
8	ABAC(XACML), XML Encryption	preserves privacy and security	-	✓	✓	✓	✓	✓	-	-	[127]

medical staff are recognized by their roles and access will be given per access policies. This approach also supports user revocation mechanisms.

2.5.3 Overview of the MLAC Model

This work also conducted research on the multi-layer access control (MLAC) mechanism, the MLAC model, to construct a secure and privacy-preserving EHR system that enables patients to share their data among stakeholders in a cloud environment. This model uses a pseudo-role attribute-based access control mechanism (PR-ABAC) which is a multi-layer mechanism that combines the advantages of both the role-based access control (RBAC) mechanism and attribute-based access control (ABAC) [124] [159]. This multi-layer access control model integrate attributes with roles combining the advantages of RBAC and ABAC and also uses the concept of provenance, aiming to ensure two fundamental security properties, the confidentiality and integrity of the sensitive data in the healthcare domain. Fig.2.8 presents an overview of the MLAC model.

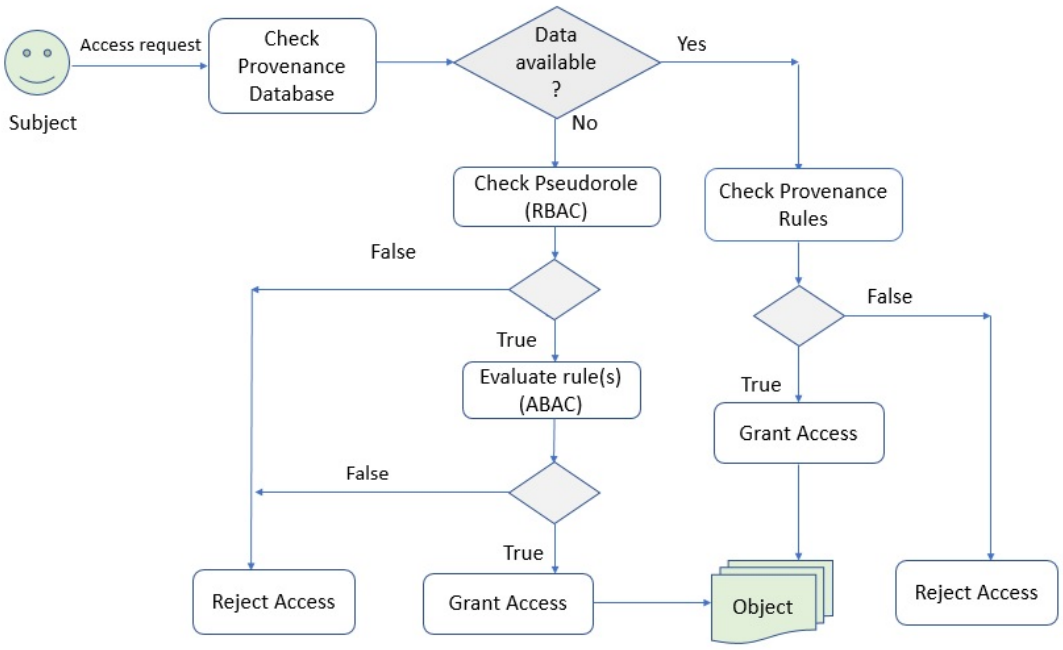


Fig. 2.8 Overview of MLAC Model.

In the MLAC model, subjects are associated with pseudoroles, which includes a set of static attributes and objects are associated with policies, which specify how attributes

are considered for access requests. When an access request is made, the policy associated with the requested object is first checked with the provenance database to see whether the corresponding data is available to grant access according to provenance rules and if not, it checks with the first layer to see whether the requester has the required pseudorole or not. If so, rules within the policies are then checked for additional fine-grained constraints (second layer) to approve or deny the access request. Fig.2.9 illustrates an XACML policy format. Initially, the policy will check whether the subject requesting access to an object holds the needed provenance rules to grant or deny access. Then the policy associated with the requested object checks to see whether the requester has the required pseudorole or not. If so, each rule is then checked to see if the access conforms to the specified values for subject, object, action and environment attributes, otherwise access is denied. This three-step process inspires the name multi-layer access control policy which permits fine-grained decisions. This work will demonstrate the applicability of the PR-RBAC model to healthcare information sharing environments.

```
< Policy >
      < ProvenanceRule > . . . < /ProvenanceRule >
      < PseudoRole > . . . < /PseudoRole >
< Rule >
      < Subject > . . .< /Subject >
      < Object > . . . < /Object >
      < Action > . . . < /Action >
      < Environment > . . . < /Environment >
< /Rule >
< /Policy >
```

Fig. 2.9 Policy Structure.

The MLAC model is described using the tuple: M = (S, O, E, A, PR, P, SPR, OP), where S is a set of subjects(users) with a predefined set of attributes SATT, that is provider, department, location etc.; O is a set of objects that are accessed by subjects with a set of attributes OATT which could be patient name, medical record number; E is the environment with a predefined set of attributes EATT which could be access time, system mode. A is a set of actions with a predefined set of attributes AATT which could be read, write, modify etc. PR is a set of pseudoroles that are composed of n attributes that are described below. P is a set of policies for fine-grained access control consisting

of two elements: a Boolean function named pseudorole, provenance rule and a set of zero or more rules. The components of the MLAC model are shown in Fig.2.10. SPR is the subject-pseudorole assignment that is a one-to-many mapping from pseudoroles to subjects and OP is the object-policy assignment relation that is a one-to-many mapping from policies to objects. A general algorithm for the MLAC model is described in Fig.2.11. This algorithm shows the ability of the MLAC model to precisely define the

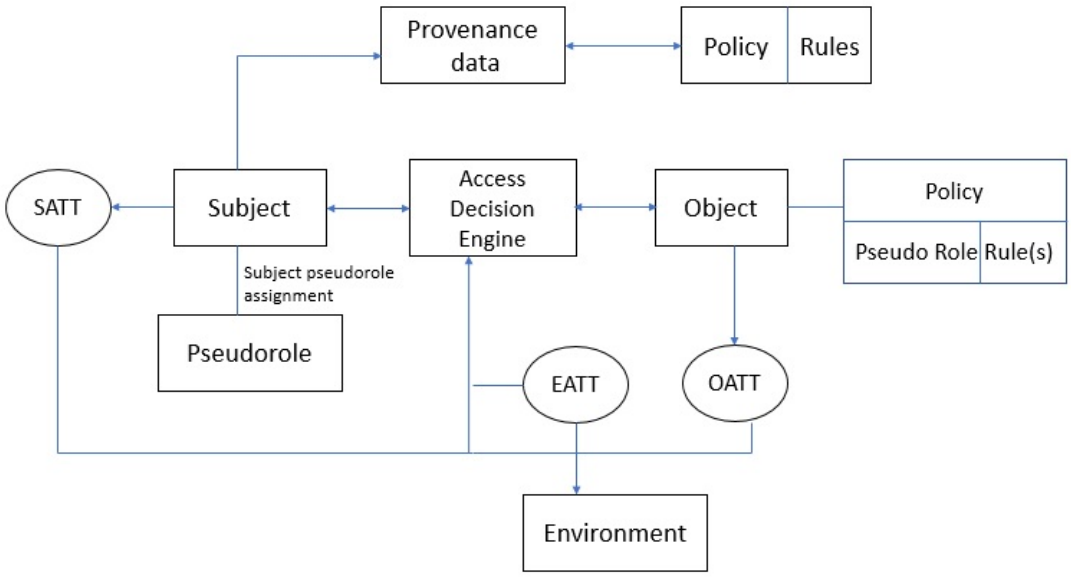


Fig. 2.10 Components of the MLAC model.

customized policies for the management of access control that uses this model. The algorithm allows or denies access to an object on the basis of the inputs that it receives. The possible inputs are object identifier (object id) i.e. the identifier of the clinical document in the EHR system, to which access is required, user identifier (subject id) is the identifier of the subject who wants to operate on the object, role indicates the role associated with the user in the EHR system, operation is the action required on the object and access mode is the mode of access such as normal and emergency mode. The output of the algorithm is PERMIT only if all the access conditions are satisfied.

This algorithm explains with which each user (clinician) is associated with a private key Sk and a common public key PK associated with the cloud server. The steps are as follows.

Input: Subject id, object id, role, operation, access mode
Output: decision $Permit, Deny$
switch (document.access mode)
Case normal:
if (AND (checkAccess(subject,object)), (policy && rules=True))
then return Permit
else return Deny
end if
break;
Case emergency:
if (AND (checkEmergency(object,role)), (policy && rules=True))
then return Permit
else return Deny
end if

Fig. 2.11 MLAC General Algorithm.

• Access Req (A_{Req}) : takes as input the identity of clinician C_{id}, the XACML query as the access structure specifying the finer attributes (τ), common public key P_k, Private key P_{rk} of the clinician, which outputs the access request (A_{Req}) = (A_{Req})(C_{id}, (τ,P_k), P_{rk}).
• Access Response(A_{Res}) : takes as input the access request AReq, the database D, public key of clinician P_k, Access Structure τ i.e. A_{Res} = A_{Res} (A_{Req}, D,(P_k,τ)).
• Response Retrieval (R_{Ret}) : takes as input the access response Ares and private key P_{rk} of the clinician and outputs the associated EHR i.e. R = R_{Ret} (A_{Res},P_{rk}).

In PR-ABAC, pseudoroles will be generated from the static attributes of subjects. Here we use the values of the attributes associated with all subjects to generate pseudoroles. Table 2.8 shows the subjects' attributes and Fig.2.12 shows how the

Table 2.8 Subjects' Attributes in the MLAC model.

Name	**ID**	**Gender**	**Provider**	**Department**	**Location**
E.Robert	345-765	Female	Physician	OB/GYN	A
A.Mark	526-874	Male	Physician	OB/GYN	A
H.John	231-938	Female	Nurse	OB/GYN	A
M. Martin	657-923	Female	Administrative Staff	OB/GYN	B

corresponding pseudoroles will be generated. Depending on the number of attributes used to generate the pseudoroles, a tree-based structure is used to identify the number of

pseudoroles. If 'n' attributes are used to generate pseudoroles, m1 x m2 x m3 x . . . x mn is the total number of generated pseudoroles where mn is the number of total different values for attribute n. However, the meaningful pseudoroles are a subset of these pseudoroles. The example in Table 2.8 uses three attributes (Provider, Department, Location) as static subject attributes that generate 18 distinct pseudoroles, as shown in Fig. 2.12.

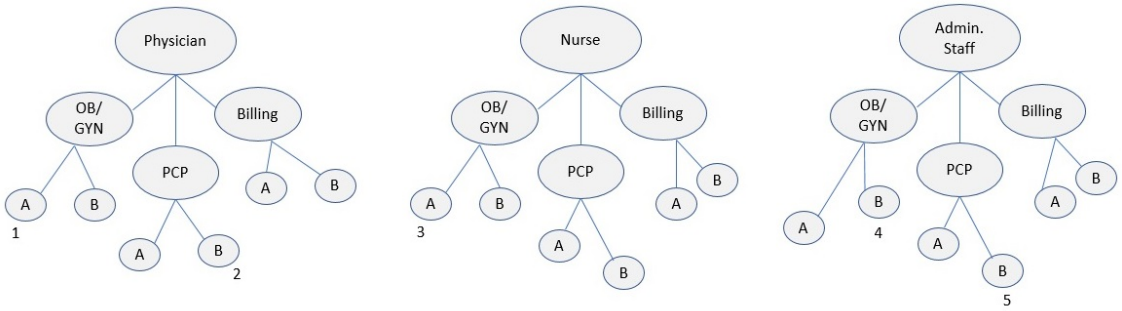


Fig. 2.12 Pseudorole Generation in the MLAC Model.

$< Policy >$
$< PseudoRole >$
$< (Subject.provider = "physician" \cup subject.provider = "nurse") \cap subject.department = \text{ɹ}any\text{ɹ} \cap subject.location = \text{ɹ}any\text{ɹ}) >$
$< /PseudoRole >$
$< Rule >$
$< Subject >$ "any" $< /Subject >$
$< Object > < object.doctorID = subject.ID > < /Object >$
$< Action > < action.type = "read" \cup action.type = "modify" > < /Action >$
$< Environment > < environment.mode = "normal" > < /Environment >$
$< /Rule >$
$< Rule >$
$< Subject >$ "any" $< /Subject >$
$< Object >$ "any" $< /Object >$
$< Action > < action.type = "read" \cap action.type = "modify" > < /Action >$
$< Environment > < environment.mode = "emergency" > < /Environment >$
$< /Rule >$
$< /Policy >$

Fig. 2.13 Policy Structure in the Clinical Section.

To preserve the privacy of patient records, a few access control rules are defined in

this use case as follows: health records are split into three sections: (1) demographic, (2) clinical, and (3) billing. An example of an access policy within the clinical section is given in Fig. 2.13. The rules are (i) subjects are not allowed to delete records in any section; (ii) physicians and nurses are allowed to read and modify records within demographic and clinical sections for patients who are under their responsibility in normal and emergency situations; (iii) physicians and nurses are allowed to read and modify records within demographic and clinical sections for non- patients in emergency situations; (iv) administrative staff are allowed to grant access to authorized users; (v) billing staff are allowed to read and modify records within the billing section. The five meaningful pseudoroles used here are: (1) Physician OB/GYN A, (2) Physician PCP B, (3) Nurse OB/GYN A, (4) Administrative Staff OB/GYN B (5) Administrative Staff PCP B. We have to define some access control rules according to the requirement of the organisation to preserve the privacy of health records. To preserve the privacy of patient records, access control rules are defined. Based on the rules and the structure of health records, some access policies are defined. To enforce the access rules, we can create an access policy accordingly such as health records within the clinical section, health records within the clinical section associated with psychiatric data with a separate access policy, health records within the demographic section which are associated with another policy, the billing section with another policy and so on according to the requirement of the organisation. However, this review is not adequate enough to protect the healthcare records from cyber attacks. Following are the research issues and challenges which have been identified by this research study.

2.6 Research Issues and Future Directions

This section discusses the research issues and future directions related to privacy and security in EHRs. Since EHR data is sensitive, confidential, and housed in third-party servers, this entails serious risks in terms of data privacy and security. Higher levels of security are critically needed to prevent, detect, and act on unauthorized access to the healthcare system which is required to mitigate social, economic, political and cultural

conflicts. Some of the main research issues are as follows:

1. How to secure and safeguard the security of stored data in the cloud?

2. How to implement privacy preserved health care data storage?

3. Which access control mechanism will be more efficient for the secure transfer of EHR?

4. Which encryption scheme can be used for preserving data security?

5. How can the health data be effectively shared against multiple healthcare providers?

6. How can the integrity of health records be maintained?

7. Who will be able to access the patient data with healthcare providers during an emergency situation?

8. What kind of access can be given to administrative staff to offset inside attacks?

9. How to handle user revocation when an authorized user leaves the system?

10.How to handle key management complexity while sharing healthcare data between disparate healthcare providers?

This review highlighted the various research issues pertaining to the privacy and security of e-health data. We found that there is an imminent need to strengthen the security infrastructure in e-health systems aiming to ensure the privacy and security of patient data by securing patient confidentiality and sovereignty. Thus, we propose several future research directions as follows:

• From the discussion, we have examined several cryptographic and non-cryptographic mechanisms. Yi et al. [156] proved that even though ABE is the most efficient of the encryption schemes, it still suffers from expensive computation costs and complexity in bi-linear pairing operations. Therefore, proposing new techniques to reduce the complexity of bi-linear operations or finding ways to outsource computations will be an interesting research direction.

• We have observed several access control mechanisms that ensure privacy in which ABAC is the most flexible and convenient for providing fine-grained access. So, ABAC will be efficient in introducing more flexibility into authorizations which can also be considered a research direction.

• Introducing secure provenance for tracking information flow for e-health data would be another interesting area on which to work.

• The integrity of health data in the cloud can be another interesting research direction.

• Privacy is a crucial aspect in healthcare. Maintaining privacy and tracking privacy violations by means of accountability mechanisms in healthcare records is essential for fraud detection and prevention. Keeping track of provenance for both data and programs is advisable.

• The great leaps in digital technologies characterised by social networking, IoT, big data analytics and cloud computing call for the immediate attention of all stakeholders to ensure stricter norms of privacy and security with respect to big data. Therefore, combinations of data analytics and artificial intelligence will be a better research focus to analyze, examine, and prevent threats in healthcare.

• A combination of encryption mechanisms and access control mechanisms to preserve big data security and privacy can also be considered as a future research direction to maintain a foolproof security mechanism in e-healthcare.

2.6.1 Discussion

From the comparative review of the existing cryptographic and non-cryptographic approaches, we have discussed how several privacy and security mechanisms can be applied to e-health data efficiently. For the comparison, we examined several crucial factors including the strengths and weakness of existing techniques and characterized each method using several privacy preserving requirements such as IN(Integrity), CO(Confidentiality), AU(Authenticity), NR(Non-repudiation), AC(Accountability), AN(Anonymity), UN(Unlinkability). The comparison results are indexed in Table 1 to Table 6 in which the symbols "✓", "✗" denote whether the specific privacy-preserving requirement is achieved or not and "−" denotes that a specific requirement is not discussed. From the detailed survey, it is evident that most of the techniques adhere to the privacy-preserving requirements but none adhere completely.

From the discussion, it is apparent that most of the existing cryptographic approaches suffer from higher computational cost, complexity in key management and distribution, in addition to vulnerability to a wide range of intruder attacks due to the

nature of design, portability and scalability. The review provides a detailed study of cryptographic approaches such as SKE, PKE, ABE, SSE, proxy re-encryption and homomorphic encryption in which SKE suffers from inflexible access control which further entails user presence for every smart card access. SKE schemes are unable to operate in a dynamically changing cloud environment because of its inability to manage multiple user roles. It is evident that PKE schemes are computationally inefficient due to larger key sizes. Even though existing ABE-based mechanisms have the advantage of defining access structures and are superior in preserving privacy levels, the computation of bilinear pairing in ABE is very expensive. One of the main limitations found in the existing techniques is that they are administered and controlled by a central trusted entity. Moreover, of the access control mechanisms, RBAC is inflexible in dynamically changing environments and the task of defining the structure and roles in RBAC is quite expensive too. ABAC is significantly efficient in handling access control, but it requires a large number of rules for decision making. The non-cryptographic approaches have several limitations in relation to their expensive processes to define and structure roles, policies, and are inefficient when operating in a dynamic environment. From the review,

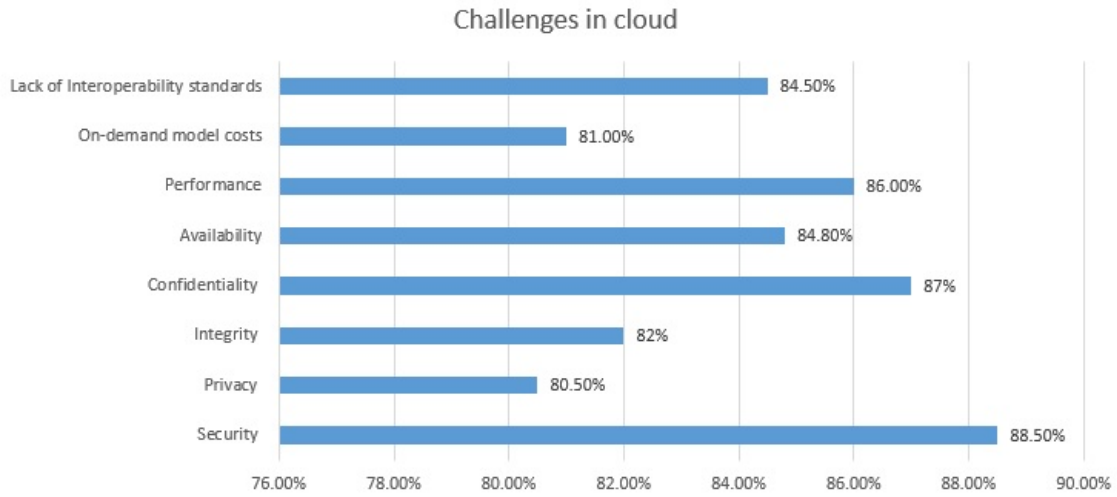


Fig. 2.14 Challenges in the Cloud.

it is also evident that SE schemes are not extensively used for handling healthcare data in the cloud environment due to computational limitations and an inability to withstand intruder attacks. The majority of the approaches described are incapable of withstanding internal and external attacks due to the lack of proper privacy-preserving mechanisms.

However, we have discussed several mechanisms and pointed out the advantages and disadvantages, but these existing techniques still fail to achieve the security, privacy and integrity of health data in e-health deployment. From Fig. 2.14, it is obvious that security is a crucial concern in the cloud environment as cyber threats are increasing exponentially. Therefore, there is an imminent need to preserve the security of EHRs against security breaches and to strengthen the security infrastructure in healthcare to ensure patient confidentiality.

One of the solutions to overcome all these limitations in the existing system is to introduce a patient -centered electronic health system namely, the Personally Controlled Electronic Health Record System, in which the patient will be the universal consent provider of their data (except in emergency situations) to all stakeholders viz doctors, pharmacists, nurses, scientists etc. Blockchain technology [162] can be used as an underlying access control tool to support this distributed ledger mechanism in the cloud. A secure blockchain-based EHR system in the cloud is depicted in Fig. 2.15. Smart

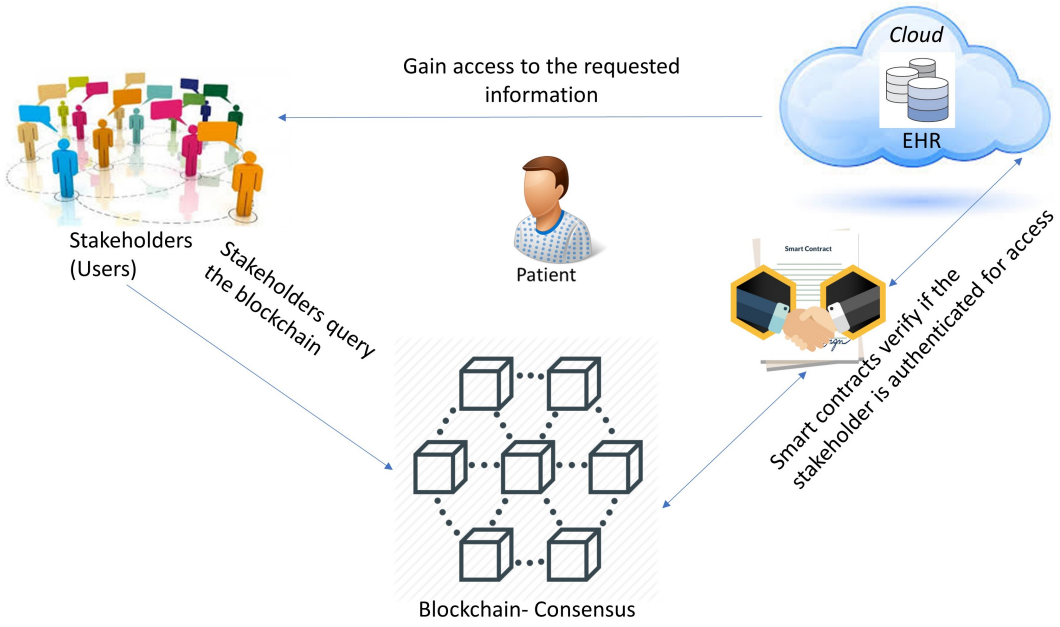


Fig. 2.15 Secure Blockchain-based EHR System in the Cloud.

contracts are intelligent permission contracts or codes that are written which verifies data ownership, permissions and the integrity of data [50]. This approach will be a

tamper-proof mechanism as every piece of health transaction information will be stored as hash values in the blockchain. It has immense potential to ensure the security, privacy, confidentiality, availability and integrity of e-health information. The introduction of this technological advancement that integrates cryptographical aspects provides a secure and efficient framework for the efficient storage, transfer and access of electronic health records in the cloud environment.

2.7 Summary

This review highlighted various research issues pertaining to the privacy and security of e-health data. As a result, we found that blockchain technology is one of the solutions to strengthen the security infrastructure in e-health systems to ensure the privacy and security of patient data and to secure patient confidentiality and sovereignty. This chapter also provides a comprehensive study of existing e-health cloud preserving cryptographic and non-cryptographic mechanisms to secure privacy aspects in the cloud and their vulnerabilities in the rapidly changing digital era.

This review presents a taxonomy of the cryptographic and non-cryptographic approaches and discussed the strengths and weakness of the existing techniques and characterized each method using several privacy- preserving requirements. In addition, this review also researched how to efficiently use the MLAC access control mechanism to determine the efficacy of data sharing among stakeholders in a cloud environment. Moreover, our work also provides and identifies key research areas with diverse aspects viz architecture, encryption techniques, access control mechanisms and has also identified some significant research issues and future research directions to ensure foolproof privacy in smart health solutions.

Chapter 3

Computational Techniques and System Description of Blockchain in Healthcare

3.1 Overview

The previous chapter discussed the cryptographic and non-cryptographic approaches in the e-health environment and their research challenges in the cloud environment. This chapter discusses the computational techniques and system description of blockchain in the healthcare ecosystem. Healthchain utilises blockchain technology to create patients' electronic health records while preserving a single true version of the user data. Distributed ledger technology or blockchain can differ in two ways: (a) the read / write access control is decentralised and not logically centralised compared with other distributed databases; and (b) it has the ability to secure transactions in competitive environments without trusted third parties.

This chapter discusses blockchain, types of blockchain, several blockchain platforms with their test bed implementations and compares various distributed storage environments. Despite the advantages provided by existing blockchain platforms, this chapter portrays the significance of Hyperledger Fabric and IPFS as the key strategies adopted to achieve the Healthchain framework's objectives.

3.2 Background: Blockchain

A blockchain-based system puts the power of patient data into the patient's hands (patient centric), and the power of data-integrity and democracy into the healthcare industry. Blockchain was used as a keyword to denote two distinct things: a data structure, a specific framework for digital information organisation and a computer system, resulting from the collaboration of a specific program on a distributed set of computers. A typical blockchain structure is shown in Fig. 3.1. Blockchain is a peer-to-peer distributed ledger technology that is linked by an incessantly growing list of records known as blocks which are secured by means of cryptographic principles by eliminating the need for a third party in monitoring asset transactions because of its underlying data non-repudiation and immutability property.

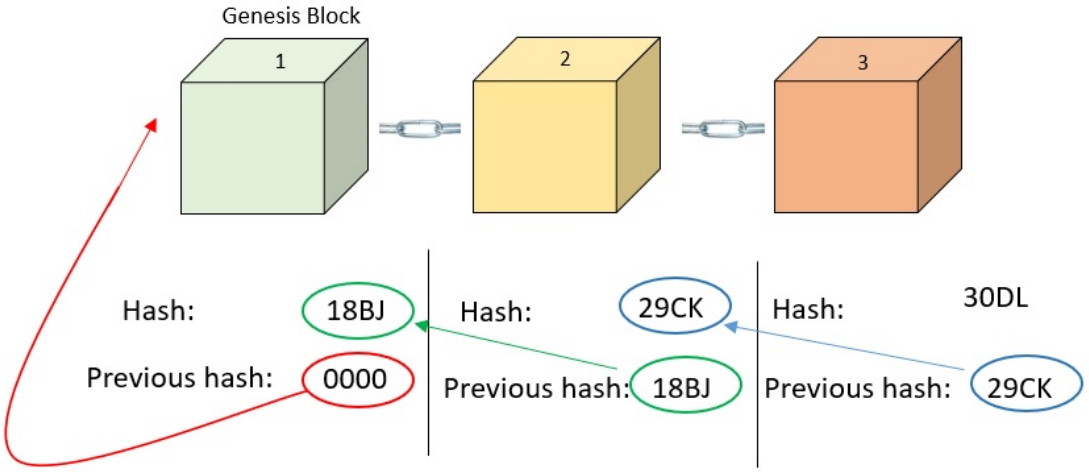


Fig. 3.1 Overview of Blockchain.

Blockchain is a public, decentralized, append-only, immutable digital ledger with a time stamped series of transactions called blocks that are linked to form a chain that is secured by means of public key encryption cryptographic principles [109] [3]. Since the blocks are linked, once the data are recorded, they cannot be altered retroactively without the modification of all subsequent blocks. A cryptographic one-way hash function (e.g. SHA-256) is also applied to the blocks to ensure immutability, anonymity and the tamper resistant structure of the blocks [134]. It can be conceptualized as a state machine running on a network of computers or nodes in which the ledger records and

stores all the transactions which occur in the network. Each peer on the network holds a full copy of the ledger which is broadcast to the peer network every time new transactions occur. Moreover, blockchain uses the consensus protocol mechanism to generate, update and validate transactions to ensure security and also employs a scripting code to run intelligent smart contracts [14].

3.3 Blockchain Models

Permissioned and permissionless blockchain systems are the two distinct types of distributed ledger network technologies that lend themselves to various types of applications and have different technical and practical implications. For permissionless or public blockchains such as Bitcoin [109] and Ethereum [44], anyone can join as a node in the network since public blockchain doesn't have any network barriers. Moreover, transactions in public chains are transparent and open though anonymity is maintained but is less desirable in the healthcare industry which manages sensitive health records. In contrast with the public blockchain, permissioned blockchain or private blockchain such as Hyperledger Fabric adopts an access control mechanism to determine the addition of a new node to the network [8].

3.3.1 Permissionless and Permissioned Blockchain

Blockchains can be public, private and consortium blockchains. In the public chain, anyone in the world can participate in the generation of blockchain and can read the contents on the chain such as Ethereum and Bitcoin. In the consortium chain, a consensus process is controlled by a set of pre-selected nodes which make up a consortium for a common purpose. In a private blockchain, right permissions are kept centralized to one organisation in which the read permissions may be public or restricted to an extent. Fig. 3.2 shows a comparison of public, consortium and private blockchains.

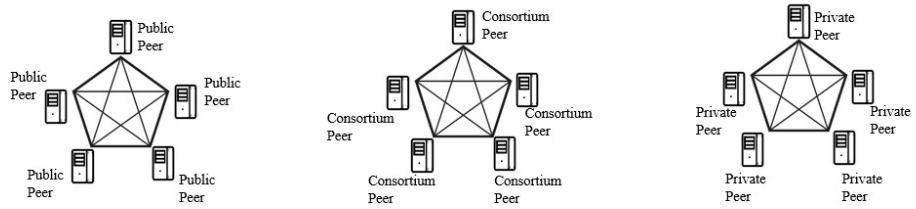
Item	Public blockchain	Consortium blockchain	Private blockchain
Topology			
Node rights	All public peers have equal rights	Some nodes have higher rights	Only private nodes have access rights
Users	All Public peer can join the P2P network	Only authorized organisation or team can join	Only authorized private peer can join and access the private P2P network

Fig. 3.2 Comparison of Different Blockchains.

3.3.2 Consensus Mechanism

This section describes consensus and the three main types of consensus associated with blockchain systems. The term consensus in this research is a mechanism that provides guaranteed ordering of transactions across network nodes and validates those transaction blocks before committing to the blockchain ledger. In a blockchain network, consensus algorithms are concerned with ensuring that the next block applied to the blockchain is a legitimate block and that all efforts to hoax participants with fake blocks from malicious or malfunctioning nodes are rejected. There are some scenarios where a distributed network may not be able to achieve consensus. Byzantine faults are one of those where it is not possible to assess whether or not a component has deteriorated. In general, a Byzantine fault tolerant (BFT) system is capable of operating as long as the number of defective nodes does not exceed one third of the total number of network nodes. In this research, BFT is implemented using Practical Byzantine Fault Tolerant (PBFT) [29] algorithm.

Proof of Work Proof of Work (PoW) is introduced for ordering the transaction and creating new blocks in the permissionless Bitcoin network [109]. Proof of work requires nodes on the network to perform a complex mathematical puzzle called mining as a way of verifying the legitimacy of transactions on this network [99]. This mathematical puzzle has a key feature asymmetry. All the network miners compete to be the first to find a solution through brute force that requires a huge number of attempts. The transactions

are placed in a block once verified and appended to the public blockchain. The difficulty of this puzzle increases proportionally to the amount of computing power in the network. i.e the more miners there are, the more difficult it is to verify the transactions. The goal is to increase a nonce such that the block's hash has a number of leading zeros significantly larger than the one needed by the system. With a greater number of zeros computation time increases exponentially and longer the chain, modification of blocks will be complex. This implies that the network is stable if it consists of a substantial number of honest nodes. However, security is only likely and the block history can be modified if the network dominates 25% of the computing power of the network is dominated by a malicious pool of nodes.

Proof of Stake Proof of Stake (PoS) is another consensus algorithm for public blockchains developed as an alternative to PoW. PoS works on the basis of the validator's possession of the stake or the computing resources in the network and not on their ability to solve a cryptographic puzzle. There are validators in the POS instead of miners. PoS takes advantage of the validators to propose, vote and create new blocks in the network. The random approach and the Byzantine fault tolerant approach are the two approaches to reach a consensus agreement [74]. A validator is randomly chosen and is allowed to add a new block in the end of the longest blockchain network. The Byzantine approach is a collective decision in which the participants are known and it is a multi-round process in which each participant needs to vote for the proposed blocks.

Consensus in Hyperledger Fabric: Practical Byzantine Fault Tolerant Practical Byzantine Fault Tolerant or PBFT consensus was introduced in 1999 for permissioned blockchain networks. The PBFT algorithm minimizes the average response time by reducing the communication overhead to run in inevitably synchronous conditions that make it ideal for Internet protocol communication systems. The PBFT algorithm involves 5 steps, namely Request, Pre-prepare, Prepare, Commit and Reply. In PBFT, nodes need to move through these stages to commit and operate in the network. PBFT will work properly even if there are some faulty nodes in the network. A request is submitted by the client to the master node in the request phase. In the pre-prepare stage,

the request is then transmitted by the master node to other nodes who decide whether to accept or deny the request. If the nodes approve the request to execute, they send a

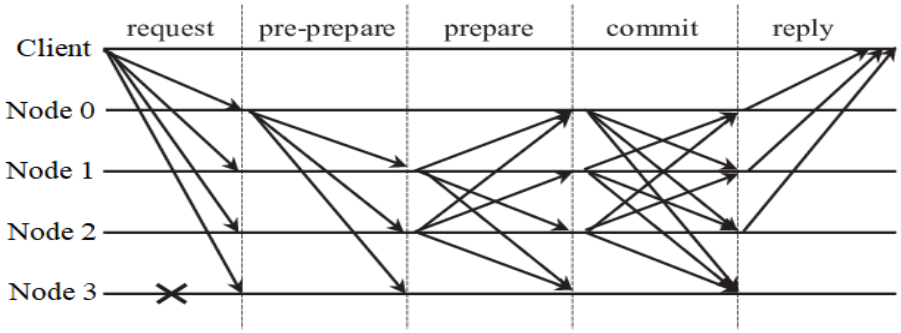


Fig. 3.3 PBFT communication (Node 3 is faulty).

prepare message to the other nodes. The nodes begin the commit stage after receiving at least 2f + 1 messages, if most of the nodes have approved the request. In the commit phase, every node sends a commit request to the other nodes and the server node finally replies to the client node in the reply phase. The client executes a timeout if any response is delayed and resends the request to the master node. It is also proven that PBFT algorithms can process up to 80,000 messages per second [19]. Researchers have developed new algorithms such as Hybrid BFT [4], XFT [93], and HoneyBadger [71] to implement efficiency and scalability.

3.3.3 Chaincode

Chaincodes are smart contracts in Hyperledger Fabric. They can be written in Go, Java and node.js. This research employs smart contracts in javascript to validate medical data entries or transactions by network participants or stakeholders. The chaincode executes in its own docker container and can be installed in peers and uses peer commands to instantiate to the channel. Within a single chain code, one or more associated smart contracts can be specified. The client application interacts with the ledger via smart contracts. There are a few default system chaincodes that govern system functionalities in Fabric namely lifecycle chaincode, query chaincode and configuration chaincode.

3.4 Blockchain Platforms and Distributed Storage: A Comparison

This section discusses various permissionless and permissioned blockchain platforms as well as several distributed data storage facilities and compares them to justify the selected methodology in the proposed framework. Here, we discuss the main blockchain platforms such as Ethereum, Ganache, Quorum and Hyperledger Fabric. Moreover we explore various decentralised data storage facilities such as Siacoin, swarm,storj and provide a detailed study of the proposed InterPlanetary File System.

3.4.1 Blockchain Platforms

Ethereum is a distributed open source permissionless blockchain network with a built-in Turing-complete programming language that can be used by anyone to build decentralised applications called Dapps via a smart contract functionality [152]. Smart contracts are written using EVM opcodes to create their own arbitrary ownership rules, transactions and state transition features which employ a secure cryptographic consensus protocol algorithm called Proof of Work. In the permissionless network, block mining takes place which is an expensive computational task for the creation of a valid block in the ledger determined by consensus. The internal fuelling mechanism in Ethereum for every transaction is ether, similar to a crytocurrency. Since the network is open, trustless and decentralized, anyone can join the network; transactions are transparent; there is a lack of trust between network nodes and network computing costs to protect against attacks towards on ledger amendments which makes the network vulnerable. In contrast to this, Ganache is used to locally set up a private Ethereum blockchain to test smart contracts designed for development and testing. Ganache supports only 10 Ethereum addresses and cannot perform network mining behaviour because it does not have miners in the network. Quorum is another Ethereum client blockchain which is improved with enterprise functionality which includes privacy features, client permission and improved performance in a private network. Despite its

strong RAFT consensus algorithm which manages faster transactions in 50ms, it still has scalability and privacy concerns in relation to applying it to a healthdata network. Hyperledger Fabric enables a green-house structure that can be modified according to the needs of the enterprise which makes it appropriate for healthcare applications.

3.4.2 Decentralised Storage

Decentralization is the transfer of authority from a central entity to a more widely distributed system. The term decentralisation is now being used in relation to Blockchain technologies such as Bitcoin and Ethereum which decentralise financial transactions and informatics. Despite the restrictive nature and data leaks of the existing storage systems, it would be ideal for storage frames to function decentrally if there were a genuinely decentralised platform. It is a system of being able to store files without having to respond to large, centralised data silos that don't undermine important values such as privacy and data security. There are several distributed storage platforms available such as Siacoin, Swarm, Storj, BigchainDB,IPFS. StorJ is an end-to-end distributed storage built on the Ethereum network and its accessing capability is restricted to the data owner and content is not available to the public which enables data to be stored in a decentralised and secure way. Swarm's primary objective is to provide an adequately decentralised and redundant store of the public record of Ethereum, in particular to store and distribute dapp code and data and blockchain data. Siacoin is also an incentive platform similar to Storj.

3.5 Hyperledger Fabric : A Permissioned Blockchain

Hyperledger Fabric is a permissioned blockchain platform developed by IBM and the Linux foundation for enterprise blockchain applications which highlights its smart contract functionality, consensus algorithm, confidentiality, scalability and resiliency. Hyperledger Fabric uses a Crash Fault Tolerant or Byzantine Fault Tolerant consensus protocol that does not require mining to achieve consensus. The Fabric architecture

	FEATURES	ETHEREUM	GANACHE	QUORUM	HLF
PLATFORMS	**PERMISSION RESTRICTIONS**	Permission less Public or Private	Permissioned	Permissioned	**Permissioned**
	DATA ACCESS	Public /private	Private	Private Ethereum based	**Private**
	PRIVACY	Existing privacy issue	Existing privacy issue	Ensures Privacy	**Privacy preserved**
	SCALABILITY	Existing scalability issue	Existing scalability issue	Scalable	**Very scalable**
	NATIVE CURRENCY	Ether	Ether	None	**Not required/can be added**
	SCRIPTING	Solidity	Solidity	Solidity	**Node.js, Go, Angular 4**
	FOCUS	General Purpose	Financial transactions	Cross-industry	**Enterprise focused**
DISTRIBUTED STORAGE	**FEATURES**	**SIACOIN**	**SWARM**	**STORJ**	**IPFS**
	STORAGE	Encrypted Cloud Storage	Storage integration with Ethereum	(Cloud's Decentralized Storage) Encrypted Data	**(DHT) Unique hash to every file**
	INCENTIVIZING MECHANISM	Incentive required	Incentive required	Incentive required	**No incentive mechanism required**

Fig. 3.4 Blockchain Platforms and Distributed Storage: Comparison.

follows an execute-order-validate paradigm aiming at resiliency, flexibility and confidentiality which gives different roles to the nodes. This can be one of the following roles: 1) Clients submit the transaction proposals for execution 2) Peers execute transaction proposals and validate transactions. All peers maintain the ledger in which all transactions are recorded in the form of a hash chain; 3) Ordering Service Nodes or Orderers that collectively form the ordering service which establishes the order of all transactions in Fabric via a consensus protocol and the endorsing peers validate transactions against an application-specific endorsement policy before committing them to the ledger [8]. Fig 3.4 shows a comparative study of different blockchain platforms and distributed storage environments. A peer node in the Fabric network stores data in blockchain and the state database as key value pairs. Peer nodes store the key value data in the state database when a peer successfully verifies a transaction. LevelDB and CouchDB are the two state database supports by Hyperledger Fabric. A peer node and its components are portrayed in Fig. 3.5. Each peer has an MSP (Membership Service Provider) for managing identities such as validate and sign endorsements for transaction

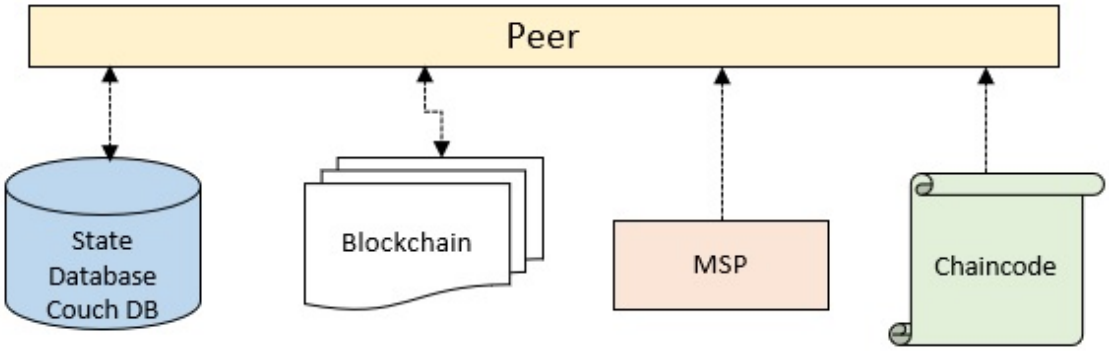


Fig. 3.5 Peer Node in Fabric Chain.

validation from peer nodes. Orderer nodes create blocks in a Fabric network in which endorsing peers execute the chaincodes and peer nodes validate the block transactions. The nodes will be submitted on the channel after validation. The transaction flow in a Hyperledger Fabric is shown in Fig 5.5. The detailed explanation is as follows:

• Initially , the client submits a transaction proposal.

•The peer node that represents the client sends the proposal to the appropriate endorsers.

•The client checks whether the proposal is endorsed correctly.

• The peer who represents the client sends the request to the ordering service.

• The ordering service orders the transactions and generates a block with a number of transactions.

• The block is then distributed to the organisation's leading peer.

• The leading peers broadcast the block on the channel to the rest of the peers.

•The peers validate block transactions before submitting them to blockchain.

3.6 System Description

3.6.1 System Overview of Hyperledger Fabric network

The application software comprises three software implementation packages:

• Hyperledger Fabric Network Package

• Hyperledger Composer Framework Package

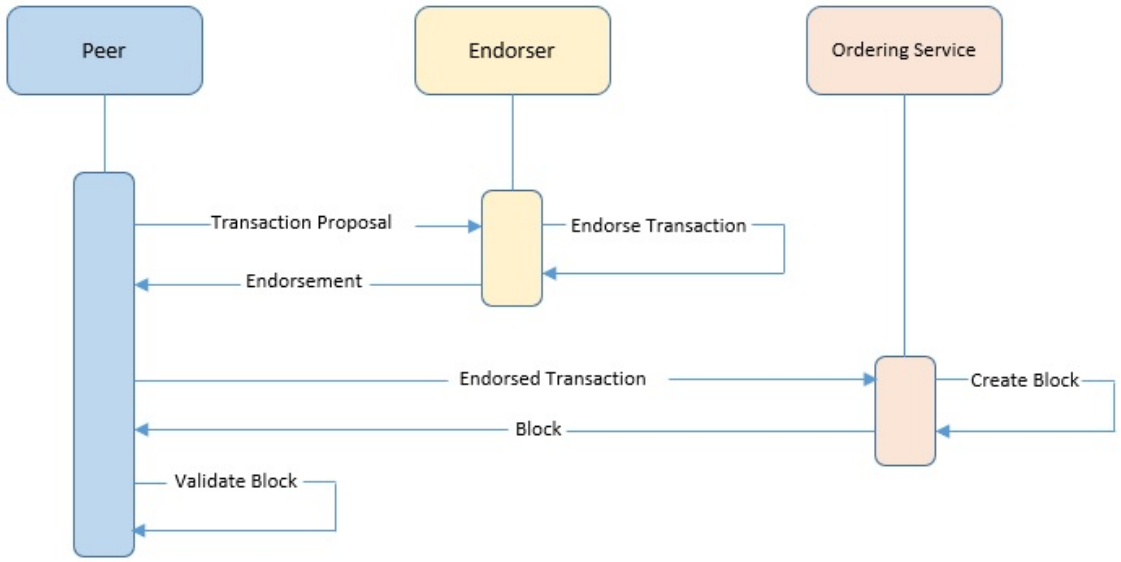


Fig. 3.6 Transaction Flow in Fabric Chain.

• Angular 4 Application Package

The HLF Network Package includes the configuration and chaincode files in the blockchain network. Executing the scripts of the package generates a functional blockchain network.The Hyperledger Composer package supports the current Hyperledger Fabric blockchain architecture and runtime, which supports plug-in blockchain consensus protocols to ensure that transactions are verified by approved participants in the business network according to policy. It comprises a Hyperledger composer modeling language (model.cto), script file that defines transaction functions (scriptfile.js), access control language (.acl) and query definitions(.qry) to define a business network. Angular4 application package demonstrates the user interface demonstration functionality associated with HLF network package in a Hyperledger Fabric blockchain environment to define the blockchain private network. Figure. 3.8 shows the directory structure of the HLF network used in this research. Chaincodes or smart contracts are written using Node.js in the proposed Healthchain network. The Update ownership, Update medical record, Doctor referral, Drug supply chain smart contracts are the chaincodes proposed in this framework.

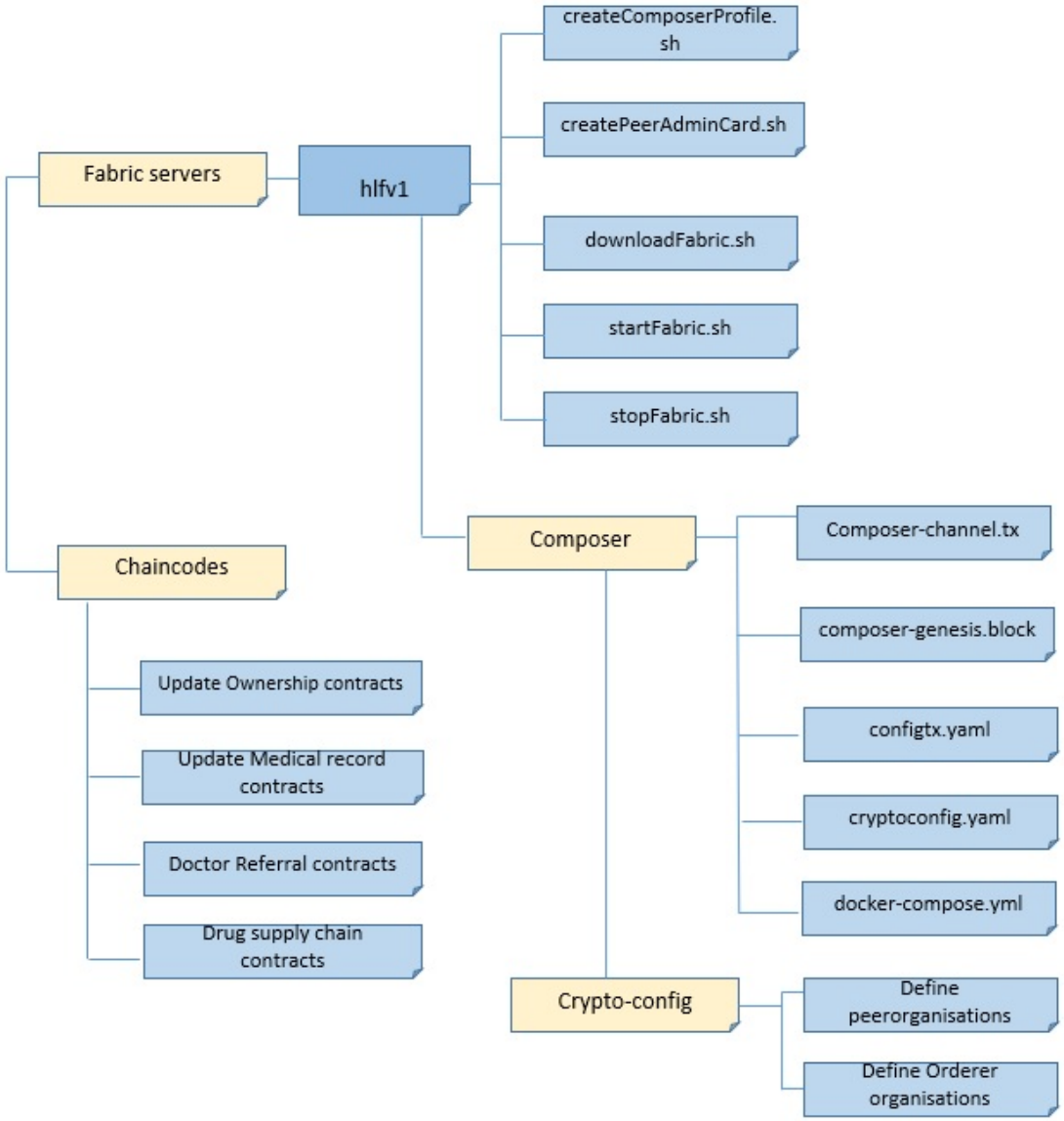


Fig. 3.7 Hyperledger Fabric Package Directory Structure.

3.7 Interplanetary File System (IPFS)

IPFS is a peer-to-peer file sharing protocol making the web better and faster, safer and more open which is appropriate for this research work. This thesis aims to contribute to scalability in blockchain by storing only the hashes of the data onchain and the actual data can be cryptographically secured and stored off the chain in decentralised storage IPFS. The main components in IPFS are as follows

• Distributed hash Tables (DHT): A data structure which has key / value pairs in the

Table 3.1 Comparative Study between IPFS and HTTP

Inter Planetary File System (IPFS)	**HyperText Transfer Protocol (HTTP)**
Decentralised peer-to-peer technique	Centralised client server approach
Get data from multiple nodes in the network hence it is copied to multiple nodes	Fails to retrieve data if server fails
Data request using cryptographic hash	Data request using IP address or domain name
Contributes historic versioning	Lifespan is 100 days
High bandwidth and retrieves data from closest peer	Low bandwidth while processing multiple requests
Resilient networks	Intermittent connections

form of a hash table. As distributed hash tables, the data is stored in a network of nodes that are efficiently organised when accessing it and the key benefits of DHTs are decentralisation, fault tolerance and scalability.

• Block Exchanges: By relying on an innovative data exchange protocol, the common file sharing programme Bitswap successfully facilitates data transfer between millions of nodes.

• Merkle DAG: Merkle DAG is a combination of a Merkle Tree and Directed Acyclic Graph which ensures that data blocks exchanged over p2p networks are precise, undamaged and unaltered by organising data blocks using cryptographic hash functions.

• Version Control Systems:IPFS uses a version control mechanism for data objects, so it is possible to access the entire file history.

• Self-certifying File System: A distributed file system that doesn't require special permissions for data exchange and uses public-key cryptography to self-certify objects by network users.

3.8 Analysis

Fig. 3.8 shows the performance throughput and scalability comparison of different blockchain platforms. Fig. 3.8(a) shows the comparison of average throughput between Hyperledger Fabric and Ethereum where 1000 transactions are deployed. Fig. 3.8(b)

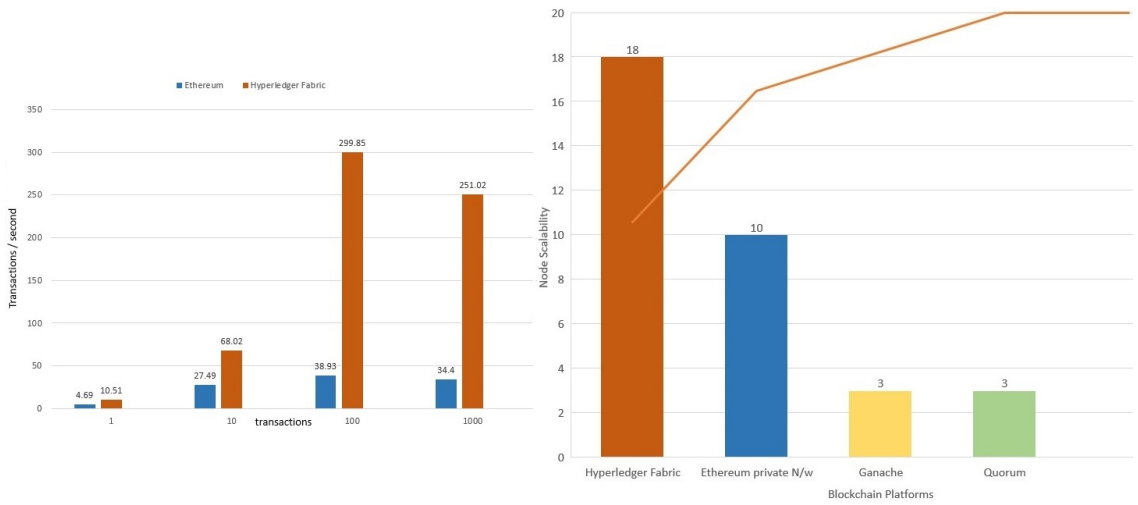


Fig. 3.8 Performance Throughput and Scalability of Different Blockchain Platforms.

compares the scalability of nodes that can be deployed in permissioned networks such as Hyperledger Fabric, Ethereum private network, Ganache and Quorum. The initial experiment conducted in IPFS was to upload the text data to determine the system feasibility. It can be seen from Fig. 3.9 that IPFS can handle the uploading of text data up to 10 MB in size and the uploading and downloading takes an average time of 6.5 milliseconds. Then the scalability is extended and analysed with huge size image uploads and downloads in the rest of this study.

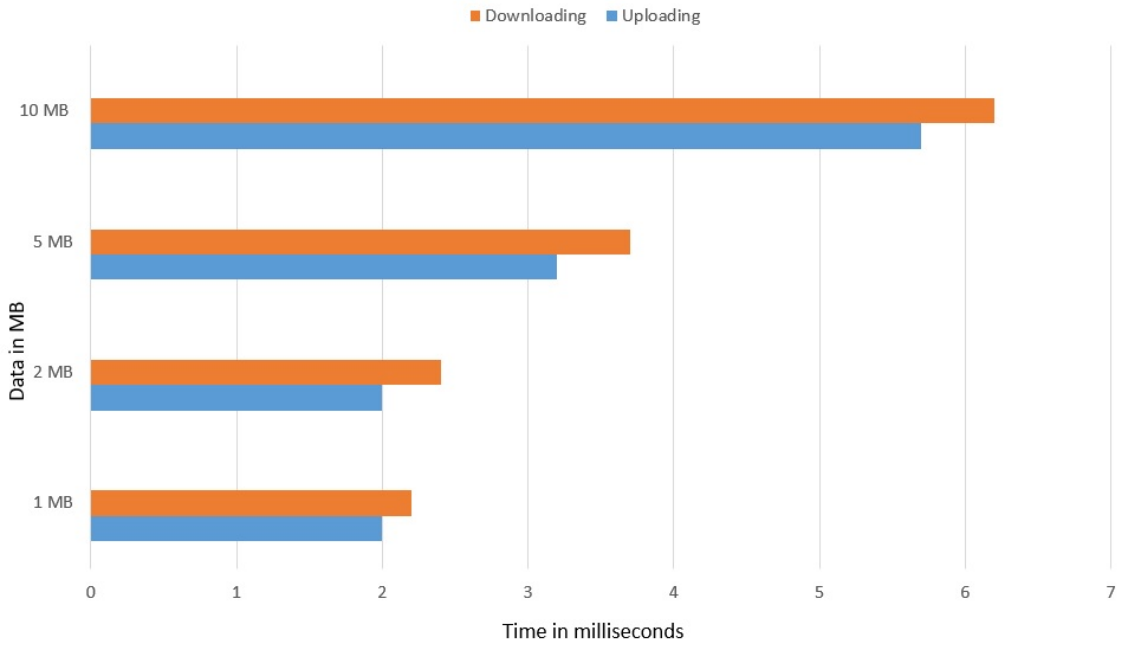


Fig. 3.9 Uploading and Downloading Time Comparison of Text Data in IPFS.

3.9 Summary

This chapter provides a brief explanation of blockchain, types of blockchain, various consensus, comparison of different blockchain platforms, system description of the proposed research and the working explanation of Hyperledger Fabric and IPFS as the key components to build the Healthchain framework. Moreover, Fig.3.4 demonstrates the significance of choosing the proposed methodological components for building the Healthchain framework. Also, an initial analysis has been carried out to portray the performance throughtput and scalability of different blockchain platforms to determine the most suitable platform for healthcare environments.

Chapter 4

Privacy Preservation Of Electronic Health Records Using Blockchain Technology: HealthChain

EHRs are stored on centralized databases in silos that increase the security risk footprint and requires trust in a single authority which makes healthcare data an extremely tempting target for attackers. Several research studies showed that centralization increases the security risk and requires trust in single authority which cannot effectively protect data from internal attacks. A lack of interoperability in EHR is another issue faced by the healthcare industry today which makes it difficult to aggregate and examine patient data, hence preventing the efficacy of EHR sharing in emergency situations. Health data in the prevalent systems is fragmented and is challenging to share with healthcare providers or stakeholders due to their varying formats and standards. Another significant concern in relation to health records housed in cloud servers is internal attacks where people with authorized credentials to access data, such as database administrators or key managers within organizations, are attackers, which is considerably worse than external attacks. Moreover, in the existing systems, patients are not in complete control of their health records since it is managed by service providers. Centralized databases can leave patients vulnerable to attacks that have escalated cyber

threats as can be seen from the recent Ransomware attack [105] and the Equifax attack [18] which hinders the privacy and security of EHRs.

This chapter describes the development of a privacy-preserving framework viz Healthchain based on blockchain technology which maintains the security, privacy, scalability and integrity of e-health data and focuses on ensuring patient privacy and data security when sharing sensitive data across the same or different organisations as well as healthcare providers in a distributed environment. The blockchain is built on Hyperledger Fabric, a permissioned distributed ledger solution using Hyperledger Composer and stores EHRs by utilizing the InterPlanetary File System (IPFS) to build this Healthchain framework. Moreover, the data stored in the IPFS is encrypted using a unique cryptographic public key encryption algorithm to create a robust blockchain solution for electronic health data. The objective is to provide a foundation for developing security solutions against cyber-attacks by exploiting the inherent features of the blockchain, and thus contributes to the robustness of healthcare information sharing environments. Through the results, the proposed model shows that the healthcare records are not traceable to unauthorized access as the model stores only the encrypted hash of the records which proves its effectiveness in terms of data security, enhanced data privacy, improved data scalability, interoperability and data integrity while sharing and accessing medical records among stakeholders across the Healthchain network.

4.1 Introduction

With the advancement in information and communication technology (ICT), most healthcare organizations have moved to electronic health records (EHRs) instead of paper-based records. EHR, electronic health data (EHD), electronic medical records (EMR) are digitalized patient records encompassing a huge variety of medical data such as medical histories, demographic information, laboratory test reports and other sensitive patient personal information including social security number and credit card information [75]. The large-scale generation and rampant usage of health information in the big data era increases the role of cloud networks not only to house the large amount

of data but also to facilitate its access across the Internet [35, 36, 88, 101]. Moreover, the lion's share of medical data is extremely sensitive and confidential, so its storage on third-party centralized servers naturally increases the privacy and security vulnerabilities that leads to several attacks including DDoS attacks [46] and Ransomware attacks which have greater ramifications beyond financial or privacy breaches [1] [25]. Considering the vulnerable nature of healthcare data in the public domain and the lack of adequate security frameworks, there is an imminent need to protect the data and devise a secure, efficient and effective mechanism to facilitate the sharing of and access to data among various stakeholders [36] [149] [147]. Blockchain technology has a large potential to bring significant efficacies to financial transactions, global supply chains, asset ledgers, healthcare and decentralized social networking.

Blockchain is one of the solutions to overcome most of the limitations in the existing distributed environment by introducing a patient-centered electronic health system namely Patient Controlled Electronic Health Record System (PCEHR), in which the patient is the universal consent provider of their data to all stakeholders except in emergency situations. Blockchain is a public, decentralized, append-only, immutable digital ledger with a time-stamped series of transactions called blocks that are linked to form a chain that is secured by means of public key encryption cryptographic principles [109] [3]. Since the blocks are linked, once the data are recorded, they cannot be altered retroactively without the modification of all subsequent blocks. A cryptographic one-way hash function (e.g. SHA-256) is also applied to the blocks to ensure immutability, anonymity and tamper-resistant structure for the blocks [134]. Moreover, blockchain uses the consensus protocol mechanism to generate, update and validate transactions to ensure security and also employs a scripting code to run intelligent smart contracts [14]. In particular, our blockchain network resolves the challenges related with interoperability, scalability, integrity, security and privacy concerns in health care data systems and delivers comprehensive clinical care. Our research exploits the inherent properties of blockchain to build a potential framework that fulfills health care use cases and supports the shift from institution-driven-interoperability to patient-centric-interoperability. This work employs Hyperledger Fabric [8] as the permissioned blockchain solution that provides a

framework for securing the interactions within the entities in the Healthchain network. The main contributions of this chapter are summarized as follows:

• Initially, this research builds a patient centric interoperability Healthchain framework in which patients will have complete control over their medical records that maintains security, privacy, scalability and integrity of the e-health data. The Healthchain framework is built on Hyperledger Fabric, a permissioned distributed ledger solutions by utilizing Hyperledger Composer and stores EHR in the InterPlanetary File System (IPFS) to build this private Healthchain network. Because of its decentralized property, this framework ensures no single point of failure and also changes to the blockchain will be visible to the participants of the Healthchain network which are immutable.

• To maintain the efficiency and scalability of the blockchain, this research stores only the hash of health records on-chain and the actual huge data is stored after encryption in the off-chain storage framework in IPFS, the decentralized storage. Furthermore, the proposed Healthchain framework only allows true records to be added on blockchain which is authenticated by consensus and access to the health records is only given based on user permission.

• The data stored in the IPFS will be encrypted using a unique public key encryption cryptographic algorithm to create robust blockchain solutions for electronic health data.

• Our research design focuses on a patient-centric approach where the patient has complete control to provide access permissions to the authorized stakeholders and does not involve any form of mining incentives beyond the efficient use of the system. This framework is a working prototype in which the blockchain technique is analyzed and also unravels the possibility of blockchain in healthcare solutions.

The remainder of the chapter is structured as follows. Section 4.2 discusses the background related work, section 4.3 discusses the preliminary components in the Healthchain framework, section 4.4 explains the cryptographical process and architecture of the proposed framework; section 4.5 details the prototype implementation of the framework; section 4.6 demonstrates the results; section 4.7 discusses the analysis and discusses of the proposed framework; and section 4.8 presents the conclusion.

4.2 Existing Techniques Using Blockchain Technology in Healthcare

This section summarizes the related works pertaining to secure storage and efficient access control schemes implemented in e-healthcare using blockchain technology. For permissionless or public blockchains such as Bitcoin [109] and Ethereum [44], anyone can join as a node in the network since public blockchain doesn't have any network barriers. Moreover, transactions in public chains are transparent and open though anonymity is maintained but this is less desirable in the healthcare industry which manages sensitive health records. In contrast with the public blockchain, permissioned blockchain or private blockchain such as Hyperledger Fabric adopts an access control mechanism to determine the addition of a new node to the network. However, the previous studies requires mining incentives in the form of ether for performing transactions in the healthcare arena.

Several tamperproof mechanisms are proposed using blockchain technology [157]. Yue et al. [160] proposed the first scheme using blockchain in the healthcare industry using a Healthcare Data Gateway, which provides patients with the ability to share their data on a private blockchain so they can manage their health data without any violation of privacy or security. However, this scheme accesses data without explicit patient agreement and does not allow other family members to allow data access in emergency situations. Also, as e-health data is growing, scalability is a major issue due to data storage on chain which further leads to the centralization of the blockchain. MedRec [11] is the first functioning prototype in healthcare based on permissionless blockchain implementation and utilizes the Ethereum smart contract functionality for the intelligent representation of medical records which are stored in individual nodes in the network. However, mining mechanisms are required to sustain the distributed ledger and also scalability is another concern with the increase of EHRs every second. Another blockchain implementation by Ivan et al. [65] is the creation of a blockchain based on EHRs in which healthcare data is encrypted and stored publicly. Another blockchain approach in healthcare is the Medchain, a permissioned network of stakeholders to

facilitate healthcare data sharing between hospitals, patients and pharmacies [129]. However, the model storing the actual data on-chain has significant privacy and scalability issues. A decentralised approach was proposed by Zyskind et al. in which the encrypted data is stored off-chain and the blockchain layer enforces access control mechanisms [170]. The Data privacy is a crucial issue with this blockchain technique as the patient's metadata is exposed, which exposes all other information. All the approaches discussed lack security, privacy and scalability and these concerns are yet to be addressed [85].

Ancile [43] is another permissionless blockchain structure which utilizes Ethereum-based smart contracts that store the hash value of the data references on blockchain for secure, interoperable and efficient access control and employs advanced cryptographic techniques such as proxy re-encryption [10] for the secure transfer of medical records. Nevertheless, Ancile has technical difficulties that include rewriting the chain structure [152], it exposes the frequency of node visits during transactions, it is unable to store huge data on chain and it incurs a high storage cost. Ancile and Medrec have scalability issues that are resolved by our framework which uses IPFS to provide secure data storage off-chain instead storing it on the chain itself. The FHIR chain proposed by Zhang et al. [164] aims at the secure sharing of clinical data by employing the Ethereum blockchain in which the onchain stores only encrypted metadata that serves as a pointer to the original healthrecords, whereas the original medical data is stored in the off chain database. Dubovitskaya et al. [48] proposed a permissioned blockchain for secure data sharing which focused on oncologic care and leverages a local database and cloud services to store the patients' encrypted data. However, this approach also makes use of an arbiter for uploading the data in the cloud which makes the system less patient-centric. Another approach proposed by Wang and Song [146] is a secure cloud-based EHR system using attribute based encryption and blockchain for the secure sharing of medical data. This approach includes the hospital as an arbiter for encrypting patients' data which again contradicts the decentralized advantage of blockchain technology and makes it less patient-centric.

There are several techniques that use blockchain technology for sharing healthcare information including EMR and PHR but still fail to address data storage and the

efficient sharing of health data [67]. Another secure cloud blockchain EHR system proposed by Wang and Song is based on an attribute-based cryptosystem integrating identity-based encryption and digital signatures[146]. Another IoT-based blockchain platform was proposed to track patients' vital signs using blockchain-based smart contracts[66]. Andrea et al. proposed a provenance management platform for tracking electronic healthcare records by employing Hyperledger Fabric blockchain smart contracts [98]. Roehrs et al.[119] presented a prototype implementation and evaluation of the OmniPHR architecture that maximizes the replication of health data across computing node models by integrating distributed health re-cords using blockchain technology and the open EHR interoperability. Another advanced decentralised privacy-preserving technique was proposed for remote patient monitoring based on the Internet of Things (IoT) technology[49].

Table 4.1 Existing Techniques Using Blockchain Technology in Healthcare.

Ref.	Addressed Challenges	Challenges to be solved
[11]	Access control, Data integrity, Interoperability	Data scalability
[129]	Data sharing , Data integrity	Data privacy , Data scalability
[43]	Access control, Interoperability, secure data transfer	Data storage
[164]	Data integrity, Access control, Interoperability	Collective decision making
[146]	Data integrity, Data security	Data storage and scalability
[67]	Interoperability, Access Control	Data Storage and Sharing
[66]	Data integrity, Global data access	Authentication, Interoperability
[98]	Interoperability, Provenance	Data storage and security
[119]	Interoperability	Scalability, Data privacy and security
[49]	Data privacy, Data security	Interoperability, Data scalability

Most of the existing approaches fail to guarantee all the essential requirements such as data privacy, security, secure storage, efficient access control, scalability and interoperability for EHRs. Our research work addresses most of the existing challenges in the e-health environment by employing a permissioned blockchain framework

utilizing Practical Byzantine Fault Tolerance(PBFT) [132] as consensus to enable data sharing in a decentralized fashion via IPFS by maintaining effective patient privacy and the confidentiality and integrity of the health records.

4.3 Components of the Healthchain Framework

A brief explanation of the preliminary components of our proposed Healthchain framework are outlined as follows:

4.3.1 Membership Service Provider

Membership Service Provider (MSP) [44] abstracts all the cryptographic mechanisms such as identity validation, signature generation and verification, protocols behind issuing and validating certificates and user authentication in the healthchain. The default interface for MSP used in this model is Fabric-Certificate Authority (CA) API and there is flexibility for the participating organizations to implement an external CA.

4.3.2 Consensus Mechanism

One key property and fundamental layer of blockchain is the consensus mechanism for transactions which depends on the smart contracts layer to validate and update transactions in the ledger in the order in which they occur. The consensus protocol determines the order of transactions and rejects bad transactions in the ledger. PBFT [8] is the consensus employed in this framework which utilizes crash fault tolerance or Byzantine fault tolerance and does not require mining to achieve consensus.

4.3.3 Hyperledger Fabric

Hyperledger Fabric [8] is the first permissioned blockchain platform that features a modular architecture established by IBM under the Linux Foundation for distributed ledger solutions. This research employs Hyperledger Fabric as the permissioned blockchain framework composed of pre-specified parties for sharing healthcare information in a reliable way without any central authority. The biggest advantages of this research in using Hyperledger Fabric is that it uses the Byzantine fault tolerant consensus protocol [104] that does not utilize mining or an associated currency to achieve consensus.

4.3.4 Couch DB

CouchD and LevelDB are the two types of peer database supports using Hyperledger Fabric. LevelDB is the default state database embedded in the peer nodes and stores chaincode data as simple key-value pairs and supports key, key range, and composite key queries. CouchDB[8] is a JSON format datastore instead of a pure key-value store that allows information mapping of the database documents. CouchDB is the on-chain database used in this research that can also improve compliance security and data protection in the healthchain.

4.3.5 Hyperledger Composer

Hyperledger Composer [45] is a set of collaborative tools for designing and modelling blockchain business networks that makes it easy and quick to build simple smart contracts and blockchain applications for business owners and developers. Composer in this research creates a business network definition comprised of model file(.cto) that defines the assets, script file(.js) with associated smart contracts, ACL(.acl) for access control rules and permissions and Query(.qry) files for defining queries to query the state database in the healthchain framework. Moreover, it packages the business

network definition to a .bna file to deploy the healthchain business network to a distributed ledger.

This model file (.cto) is written in an object-oriented language known as the Hyperledger Composer Modeling Language. In Healthchain, the model file defines the components and resources required for the blockchain network that includes namespace, resources and imports. The resources include assets, participants, transactions, enumerated types, and events required for the blockchain network. The model file in this research (org.ehr.healthchain.cto) is given in Appendix A.1. Queries provide a WHERE clause which specifies the parameters that are used to choose assets or participants in the network. The query (.qry) file used in this research is given in Appendix A.3.

4.3.6 SmartContracts- Chaincode

Smart contracts are self- executing chain codes that encode the rules of certain network transactions and are currently written in the Go language that is installed and instanced by authorized participants on channel peers. This research work uses smart contracts that encompass the application logic of the system for EHR transactions particularly for data transmission, access management, and request handling such as updating medical records, allowing doctors to write, e-referrals to other doctors, updating ownerships, sending e-prescriptions to pharmacists. Smart contracts will be executed during user interaction to identify requests, validate requests, grant access permissions, and update permissions for medical records. A snippet of the smart contract used in this research is given Appendix A.2.

4.3.7 Interplanetary File System

IPFS [17] is a peer-to-peer distributed file system that shifts the present version of the web to a distributed version and it can be used to replace HTTP. For example, if we want to retrieve a data structure or download a file that is available on the web using IPFS, it can be retrieved through the peers in the network using a cryptographic hash

or unique fingerprint of that file using the content addressing property of IPFS. IPFS stores the encrypted data in multiple nodes if the data is higher than a defined threshold (size>256KB). In the context of this research, IPFS is used as an off-chain database for the storage of infinite healthcare records in which the medical records are encrypted using public key encryption before storage and the hash of the health records will be stored in the couch database.

4.4 Proposed Methodological Framework

The proposed Healthchain architecture is shown in Fig. 4.1. This framework includes Angular 4, Composer Rest Server, Hyperledger Composer, Hyperledger Fabric, Chaincode, CouchDB, IPFS and Fabric Client. Angular 4 is the front end of the Dapp

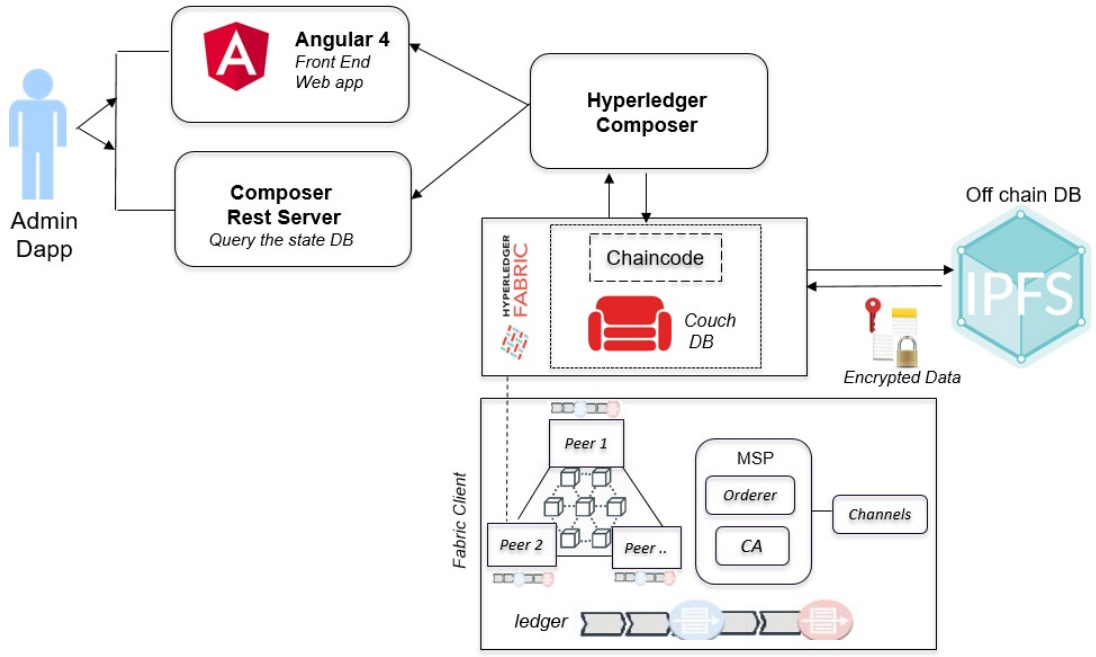


Fig. 4.1 Healthchain Architecture.

(decentralized application) framework that connects with the Composer Rest server that exposes and visualizes the state database, couchDB . The Dapp admin interacts with the user interface via the Angular framework and the application processes user requests to the fabric network through a REST API known as the composer Rest Server. The REST

API is used to retrieve the current state of the on-chain database which is the couchDB where the Angular framework retrieves the data through GET calls to the composer Rest API. Hyperledger Composer builds and models the blockchain business network to create smart contracts for decentralized applications. Hyperledger Fabric [8] is the permissioned blockchain platform for distributed ledger solutions that supports the development of smart contracts known as chaincodes which are writable in Go, Java and Node.js to validate medical data entries by network participants.

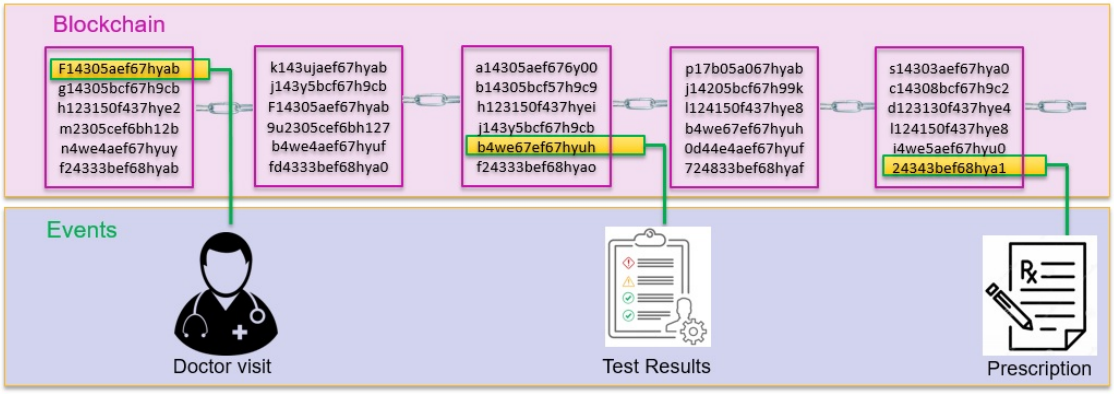


Fig. 4.2 Overview of Healthchain.

The Healthchain framework employs a two-pronged solution platform (1) an on-chain solution implemented on the secure network of Hyperledger Fabric which utilizes the on-chain database Couch DB; and (2) an off-chain solution to securely store data via IPFS (Interplanetary File System). Similar to Bitcoin [109] which is designed to maintain financial transactions, Healthchain is intended for transactions in healthcare that are secured via cryptography. In Healthchain, any interactions with the health records will be recorded as a transaction on the network and the transactions will be visible only to the participants related to the transaction. An overview of the Healthchain is shown in Fig.4.2. It shows a log of transactions as hash values in the blockchain for every event which occurs in healthcare such as record creation, access, modification or updating. From Fig. 2.15, it is evident that each transaction has a unique hash that guarantees the integrity of the health records and allows only append-only revisions. Moreover, it produces a different hash which will not match the prior hash if the record has been tampered with. When the identity management is combined with blockchain applications, the ledger becomes the supreme indicator of who did what and

when on a blockchain.

The working prototype is implemented on a permissioned blockchain called HealthChain on Hyperledger Fabric by employing Hyperledger Composer to create decentralized web applications for a single organization by incorporating three peer nodes as shown in Fig.4.3. This organization has three peer nodes with one anchor peer node as the validating node and an ordering node (Kafka) with a single public channel for registering the network participants. The system contains multiple peer nodes

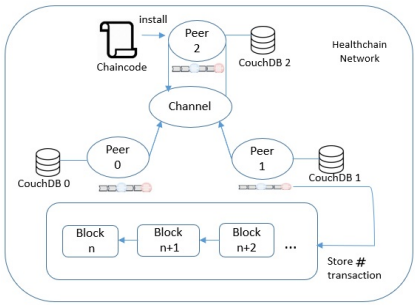


Fig. 4.3 Nodes in Healthchain.

configured to use corresponding CouchDB as the world state database and IPFS as the distributed database, a solo ordering node, a Certificate Authority, Membership Service Provider (MSP) and smart contracts to connect to the blockchain. This can be extended to multiple peer nodes and multiple organisations in different machines to prove the system's scalability. This framework has ledgers and associated smart contracts which has access to the ledgers. The application connects with peer nodes that invokes smart contracts to update the ledger.

The Hyperledger Fabric healthchain network is built in a single organisation with three peer nodes using docker containers on the local computer but clearly, in the real world, it would be in separate IP networks or protected cloud environments. The organisation's three peers are labelled peer0 (P0), peer1(P1) and peer2 (P2) in which each holds their own instance of ledgers and copies of smart contracts. A single channel is designed so that the Hyperledger Composer can communicate with peers via the channel. In this network, our application A1 generates a transaction T1 to peer0, peer1 and peer2 via channel C. Whenever a transaction is executed, the chaincode will be installed to the peers. The application interacts with peers and invokes chaincodes for

querying or modifying the ledger. The transactions are stored within the blocks as hash values in the blockchain which enables the history of changes that contributed to the healthchain framework. A block in the ledger pertaining to the health record of a patient i mainly comprises the workload of that transaction $W_t(i)$, hash of the previous transaction $Wp_{\#}(i)$ and hash of the current transaction $W_{\#}(i)$. The total workload of that block can be calculated as $W_{Tot}(i)$:

$$W_{Tot}(i) = W_t(i) + Wp_{\#}(i) + W_{\#}(i) \tag{4.1}$$

4.4.1 Cryptographical Process in HealthChain

Blockchain systems leverage cryptographic techniques to ensure data integrity and confidentiality. This research employs special public key cryptography to encrypt the data in the off chain storage, IPFS. The wholistic view of the patient-doctor interaction for accessing health records is outlined as shown in Fig.4.4. The clinician (Doctor)

Table 4.2 Explanation of Notations.

Notations	Definition
IPFS	InterPlanetary File System
P_{Cv}	Composite data view
S_k	Session Key
C_{Pk}	Public Key of Clinician
C_{Pr}	Private Key of Clinician
P_{EHR}	Patients' Health record
P_{Pk}	Patients' public key
P_{Pr}	Patients' private key
P_i	Patient
C_i	Clinician
R_i	Receptionist
Ph_i	Pharmacist
UP_{Cv}	Updated Composite view
UP_{EHR}	Updated Health Record

requests permission to access the health record of the patient stored in the IPFS. The patient approves or grants the request of permissioned users on the basis of role and rule-based access control permissions, as shown in Fig.4.5 and Fig.4.6. The system in

this framework refers to the client-side application. The system generates a composite view of the record on the basis of the request, alternately sharing the whole patient data. The system further generates a session key S_k to access records for a definite session and encrypts the composite view with the session key and then stores it in IPFS. The system will also send the encrypted session key and encrypted composite view to the clinician. Furthermore, the system also shares the encrypted session key with the patient. The clinician decrypts the session key, decrypts the composite view and updates the composite view as the updated record. Further, the clinician resolves the instance after encrypting the updated record with the session key and uploads it to the IPFS. The system notifies the patient of the record updates. The system decrypts the updated composite view using the session key and decrypts the encrypted medical record with the patient's private key from the IPFS. Finally, the system commits the updates to the original record, encrypts the original record with the public key of the patient and uploads it to the IPFS. The session key and the composite view for each session expires on session completion. The procedure can be explained with a detailed notation in the following algorithms:

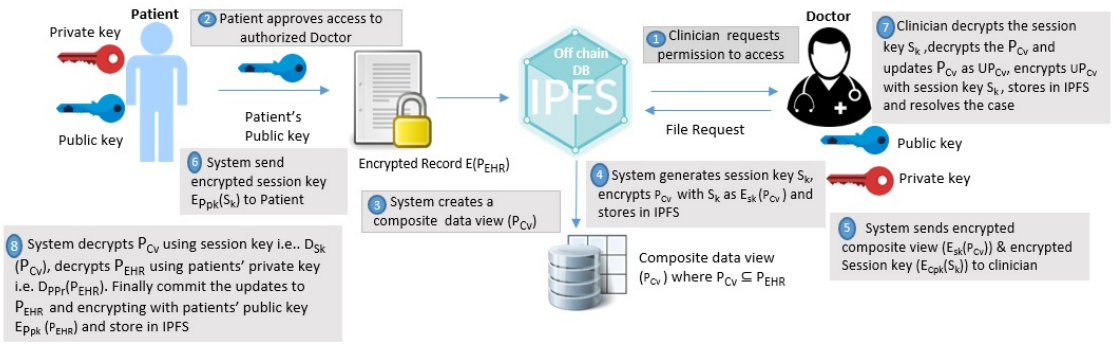


Fig. 4.4 Cryptographical process in Healthchain.

4.4.2 Proposed Algorithms

Table 6.1 explains the notations used in the algorithms and Algorithm 1 is used by the clinician to create and update the health records in the healthchain. In our Healthchain framework there are four stakeholders, where P is Patient, C is Clinician, R is

Receptionist and Ph is Pharmacist. We assume there are n participants for each stakeholder in the proposed framework. The Fabric-CA issues public key certificates to all n participants such as Patient, Clinician, Receptionist and Pharmacist. There will be a key pair for each participant in which P_{Pk_i} and P_{Pr_i} are the public and private keys of the patient P_i, C_{Pk_i} and C_{Pr_i} are the public and private keys of clinician C_i , R_{Pk_i} and R_{Pr_i} are the public and private keys of the Receptionist R_i and Ph_{Pk_i} and Ph_{Pr_i} as the public and private keys of the Pharmacist Ph_i respectively where *i*=1 to *n*. This scenario gives a detailed explanation of how the Clinician and Patient interact to access health records in the Healthchain framework. Algorithm 1 is explained as follows. Consider that patient P_i grants access to clinician C_i to his/her medical record P_{EHR_i} upon request based on the access control permissions as shown in Fig .4.5 and Fig .4.6. The system then creates a composite view P_{Cv_i} of the patient record P_{EHR_i} that is accessible to clinician C_i on request, alternately sharing the whole medical records of the patient. Composite view P_{Cv_i} is the attribute set of the stored medical record P_{EHR_i} that the system creates on the permissioned user request without sharing the complete patient record. The composite view of a specific health record restricts access to the original data in such a way that a user can see and modify only the selected data they need and no more. In other words P_{Cv_i} is a subset of P_{EHR_i} as shown in equation (4.2) and (4.3).

$$P_{Cv_i} \subseteq P_{EHR_i} \tag{4.2}$$

$$P_{Cv_i} = (D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i}))) \tag{4.3}$$

The system further generates a session key S_k shared between the clinician and the patient for a definite session. The system then sends the encrypted session key S_k to the patient as $E_{P_{pk_i}}(S_k)$ and clinician as $E_{C_{pk_i}}(S_k)$ by encrypting using respective public keys of the patient P_{Pk_i} and clinician C_{Pk_i} for a distinct session as shown in step (8) and step (9) in Algorithm 1. The composite view P_{Cv_i} will also be encrypted with session key S_k as $E_{S_k}(P_{Cv_i})$ and stores it in IPFS. In addition, the system sends encrypted composite view $E_{S_k}(P_{Cv_i})$ to the clinician. Here Algorithm 1 calls Algorithm 2 for the clinician's update of the health records. Now, the clinician decrypts the session key with

Algorithm 1 System():Create and update composite view of Medical Records

Input: A Clinician C_i with public key C_{Pk_i} and session key S_k to access medical record P_{EHR_i}

Output: Creation and update of the medical record

procedure Electronic Health Record (P_{EHR_i})
for each user U with access permission to P_{EHR_i}
Algorithm checks xml access permission rules to grant or deny access to the user
if (permission type ==“ALLOW” && role Type==‘Clinician’) **then**
    Create composite view P_{Cv_i} of the medical record P_{EHR_i} in IPFS
    $P_{Cv_i} \rightarrow \int_{i=1}^{n} (D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i})))$
    $P_{Cv_i} \subseteq P_{EHR_i}$
    Generate a session key S_k
    $P_i \leftarrow E_{P_{Pk_i}}(S_k)$ /*Send encrypted session key to patient
    $C_i \leftarrow E_{C_{Pk_i}}(S_k)$ /*Send encrypted session key to clinician
    $C_i \leftarrow E_{S_k}(P_{Cv_i})$ /* Send encrypted composite view to clinician
    Algorithm 2() /* Call Algorithm 2() for clinician record access and update
    $P_{EHR_i} \leftarrow (D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i})))$
    $UP_{Cv_i} \leftarrow (D_{S_k}(E_{S_k}(UP_{Cv_i})))$
    $P_{EHR_i} \leftarrow [(D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i}))) + E_{P_{Pk_i}}(UP_{Cv_i})]$ /*System commits the update to the original record
    return #
else
    access $\leftarrow$ deny
end if
return access
end procedure

Algorithm 2 ():Algorithm for clinician creating and updating medical records in Healthchain

Input: A Clinician C_i with public key C_{Pk_i} and session key S_k to create medical record P_{EHR_i}

Output: Record Creation and updation

procedure Clinician (C_{Pk_i})
for each clinician with access permission on receiving encrypted S_k and P_{Cv_i}
$C_i \leftarrow D_{C_{Pr_i}}(S_k)$ /*Decrypt session key with Clinician’s private key
$C_i \leftarrow D_{S_k}(P_{Cv_i})$ /*Decrypt composite view with clinician’s session key
$P_{Cv_i} \rightarrow (UP_{Cv_i})$ /* Clinician updates Composite view
IPFS $\leftarrow E_{S_k}(UP_{Cv_i})$ /* Encrypts updated composite view with Clinician’s session key
System() /*call System()
end procedure

his private key and decrypts the composite view with the session key as shown in step (2) and step (3) in Algorithm 2. If there are any updates, the clinician updates P_{Cv_i} as UP_{Cv_i}, resolves the case, encrypts with the session key and uploads UP_{Cv_i} to IPFS as E_{S_k}(UP_{Cv_i}). The system refers to the client-side application in this framework. The patient uses a pass code to encrypt the private key P$_{Pr_i}$ and stores it on the client side. For convenience, the patient can provide this pass code that decrypts the private key every time instead of sharing or uploading the private key, and the client end application can use this private key to decrypt the medical record. On the clinician's record modification, the update calls Algorithm 1 in which the system decrypts the encrypted record ie. $E_{P_{Pk_i}}(P_{EHR_i})$ using the patient's private key and decrypts the encrypted updated composite view from the IPFS ie. E_{S_k}(UP_{Cv_i}) using the session key as shown in steps (12) and step (13) in Algorithm 1. Finally, the patient commits the updates to the original record and encrypts the original record P_{EHR_i} as $E_{P_{Pk_i}}(P_{EHR_i})$ before uploading it to IPFS as shown in equation (4.4).

$$P_{EHR_i} = [(D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i}))) + (E_{P_{Pk_i}}(U_{P_{Cv_i}}))] \tag{4.4}$$

The session key S$_k$ and the composite view P_{Cv_i} for each session expires on session completion. The transactions eventuated on the clinician access and record updates that invoke smart contracts thus creates a unique hash value and is added to the Healthchain. This is composed of two algorithms, Algorithm 1 and Algorithm 2 .

4.4.2.1 Access Control Permission Rules

Fig. 4.5 and Fig. 4.6 shows a snippet of the XML structure of access control permission rules in the Healthchain network. Algorithm 1 checks the access management rules in Fig. 4.5 and Fig. 4.6 for granting or denying access to the health records. Access control policies are designed to safeguard the privacy of patients' healthcare records. Fig. 2 presents the algorithm for the clinician to create and update the health records in the Healthchain network. When an access request is made, the algorithm verifies the access control rules that are written in extensible markup language in Fig. 4.5 and Fig. 4.6 which

```
<?xml version="1.0"?>
<access-control-rules>
    <role name> Clinician Ci </role_name>
    <permissions desc= "Permissioned Clinician authorized by the Patient">
    <resource desc= "EHRi"> Electronic Health Record </resource>
        <Object>
            <object.ownershipid=Ci.id>
            </Object>
            <action type="read"> ALLOW </action>
            <access mode>
                <access. mode=“normal”>
                </access mode>
            </permissions>
            <role name> Clinician Ci </role_name>
             <permissions desc= "Permissioned Clinician authorized by the Patient
             and can modify the record for a particular session">
    <resource desc= "EHRi"> Electronic Health Record </resource>
        <Object>
            <object.ownershipid=Ci.id>
            </Object>
            <action type="write"> ALLOW </action>
            <access mode>
                <access. mode=“normal”>
                </access mode>
            </permissions>
             ......
             <role name> Clinician Ci </role_name>
             <permissions desc= "Permissionless Clinician">
    <resource desc= "EHRi"> Electronic Health Record </resource>
        <Object>
            <object.ownershipid≠Ci.id>
            </Object>
            <action type="read">DENY</action>
            <access mode>
                <access. mode=“normal”>
                </access mode>
            </permissions>
             ......
             <role name> Pharmacist Phi </role_name>
            <permissions desc= "Permissioned user authorized by the Patient for
            a particular time period">
```

Fig. 4.5 A Snippet of the XML Document Showing Access Control Permission Rules.

defines the access rights of the user on resource EHR_i defined by the owner. These access rules will be stored in the blockchain and submitted to the blockchain channel through a transaction called the Business Network Archive Transaction. In this approach:

–the rules comprise the condition specifying the ID of the subject to which the access control policy grants the right of access;

–conditions specifying sets of values are authorized for the subject, resource, action type and environment attributes for access to be granted.

In our framework, we designed the rules to modify these conditions when they transfer these access rights to other authorised users before submitting the data to the healthchain.

```
    <access. mode="normal">
    </access mode>
</permissions>
 ......
 <role name> Pharmacist Phi </role_name>
<permissions desc= "Permissioned user authorized by the Patient for
a particular time period">
<Object>
<object.ownershipid=Phi.id>
</Object>
<action type="read"> ALLOW </action>
<access mode>
    <access. mode="normal OR emergency">
    </access mode>
     </permissions>
     <role name> Referred Clinician Ci </role_name>
<permissions desc= "Permissioned referred clinician authorized by the Patient for
a particular session">
<Object>
<object.ownershipid=Ci.id>
</Object>
<action type="create/update"> ALLOW </action>
<access mode>
    <access. mode="normal OR emergency">
    </access mode>
     </permissions>
      ......
      <role name> Receptionist Ri </role_name>
<permissions desc= "Permissioned user authorized by the Patient for
a particular time period">
<Object>
<object.ownershipid=Ri.id>
</Object>
<action type="view"> ALLOW </action>
<access mode>
    <access. mode="normal OR emergency">
    </access mode>
     </permissions>
      ......
       </access-control-rules>
```

Fig. 4.6 Access Control Permission Rules for Healthchain Network.

The actors in this scenario are resource owner P, Resource EHR_i and several subjects such as C_i, Ph_i and R_i in the healthchain framework. The clinician C_i or any user can only read, write, modify or update access to health records according to the access control permissions. From Fig. 4.5, it is clear that if the subject ID matches with the object ownership ID and only if the subject is a permissioned stakeholder, permissions such as read, write access are allowed or otherwise access will be denied. Stakeholders such as Pharmacist and Receptionist in this healthchain framework have given read access only to the composite view of the health records for a particular session if their subject ID matches the ownership ID of the object or resource as shown in Fig. 4.6 .

4.5 Prototype Implementation of the Proposed Framework

This section gives a detailed description of how users and records are added in the healthchain framework, the steps included to provide access permissions to authorized users and the retrieval of records in the healthchain framework.

4.5.1 Adding Users to the Healthchain Framework

The process for adding users to the healthchain network can be seen in Fig. 4.7. The framework developed is role-based in which Patients, Clinicians(Doctors), Chemists and Receptionists can register themselves and login using login credentials such as email address and password. The nodes will be added by the network admin to the blockchain after validation from the consensus voter nodes. The patients' and the users'

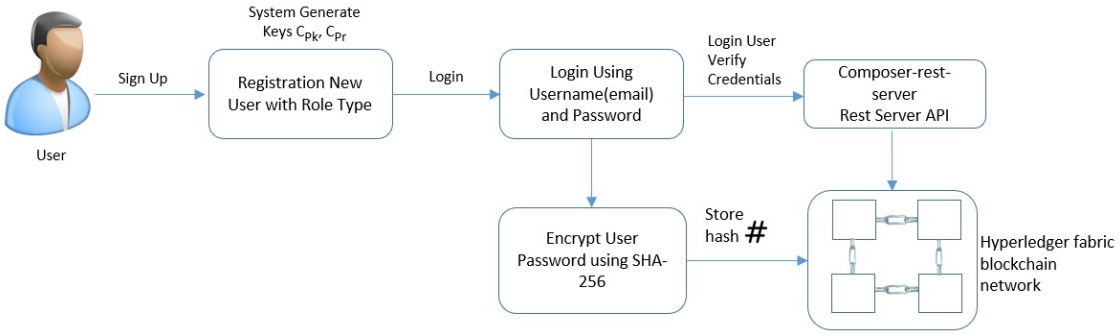


Fig. 4.7 Adding Users to Healthchain.

will be added to the healthchain with limited validation using their credentials such as username and password with each user having public private key pairs Pk_i, Pr_i. The user password is encrypted using the SHA-256 hashing algorithm for improved security. The Composer Rest Server generates a REST API from the deployed blockchain business network that visualizes and queries the values stored in the couch database. The rest server also performs create, read, update and delete operations for assets and participants which allows transactions for processing and retrieval. A snippet of the component file used in this research is shown in Appendix. B.1. This Component.ts file

defines the class, module, properties and processing of the corresponding html component to add users to the blockchain network. The processing includes connecting to the database, interacting with other components, routing etc.

4.5.2 Adding Records to the Healthchain Framework

The stage-by-stage explanation in Fig. 4.8 shows the medical record added for the patient by the clinician to the healthchain. This approach begins with assuming that the patient and the clinician have established an authorized relationship for updating health records. The process of adding medical records to the database by the clinician is via the

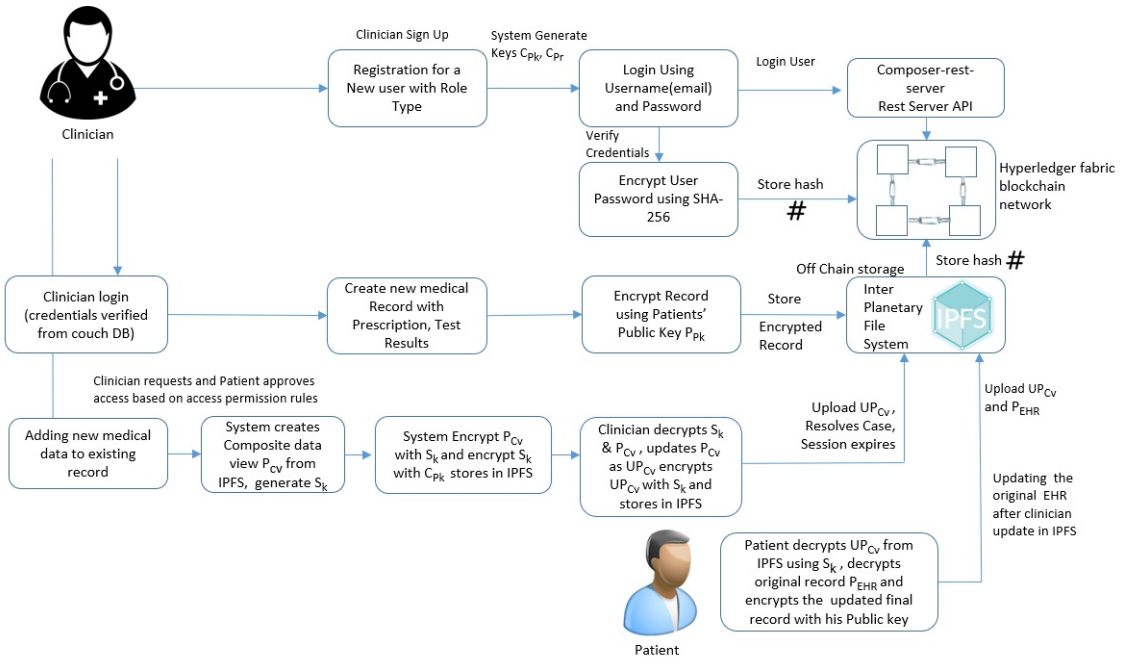


Fig. 4.8 Adding Records to Healthchain.

internal encryption mechanism. There are two scenarios on adding patient records to the healthchain: (a) A new patient record will be created by the clinician to the healthchain by uploading the encrypted medical record using the patients' public key to the IPFS; (b) A new patient record will be added or modified by the clinician and the system creates a composite view, P_{Cv_i} of the data that can be accessible to the clinician C_i alternately sharing the whole data. The system further generates a session key S_k shared by the patient and the clinician for a distinct session. The system then sends the

encrypted session key S_k to the patient as $E_{P_{pk_i}}(S_k)$ and the clinician as $E_{C_{pk_i}}(S_k)$ by encryption using the respective public keys of the patient P_{Pk_i} and clinician C_{Pk_i} for a distinct session. The Composite view P_{Cv_i} will also be encrypted with session key S_k as $E_{S_k}(P_{Cv_i})$ and stores it in IPFS. In addition, the system sends the encrypted Composite view i.e.$E_{S_k}(P_{Cv_i})$ to the clinician.

Now, the clinician decrypts the session key with his private key and decrypts the composite view with the session key. If there are any updates, the clinician updates P_{Cv_i} as UP_{Cv_i}, resolves the case, encrypts with the session key and uploads UP_{Cv_i} to IPFS as $E_{S_k}(UP_{Cv_i})$. On the clinicians' record update, the system decrypts the encrypted record i.e. $E_{P_{Pk_i}}(P_{EHR_i})$ using the patients' private key and also decrypts the encrypted updated composite view from the IPFS i.e. $E_{S_k}(UP_{Cv_i})$ using the session key. Finally, the patient commits the updates to the original record and encrypts the original record P_{EHR_i} as $E_{P_{Pk_i}}(P_{EHR_i})$ before uploading to IPFS. The session key S_k for each session expires and the composite view P_{Cv_i} will be deleted after the session is completed. The transactions eventuated on clinician access and record updates will be hashed by employing smart contracts and added to the healthchain. This procedure is summarized by Algorithm 1 and Algorithm 2 by employing Fig. 4.5 and Fig. 4.6 for access management. A snippet of the component file used for adding records to the healthchain is shown in Appendix B.2.

4.5.3 Providing Access Permissions to Authorized Users

The patient has complete control and ownership to grant read, write, deny or revoke access permissions to the provider or other stakeholders such as the receptionist, doctor or pharmacist on the medical record thereby maintaining restrictive access control. The XML rules shown in Fig. 4.5 and Fig. 4.6 present the read, write and deny access permission rules in the proposed healthchain. Moreover, the patient can permit access to health records based on the authenticated user approved by the consensus in accordance with role type and permission type. Furthermore, the patient can also revoke the access of a particular clinician to his medical records and in that situation, permission to further access the record can be denied. As shown in Fig. 4.9, Healthchain uses permission

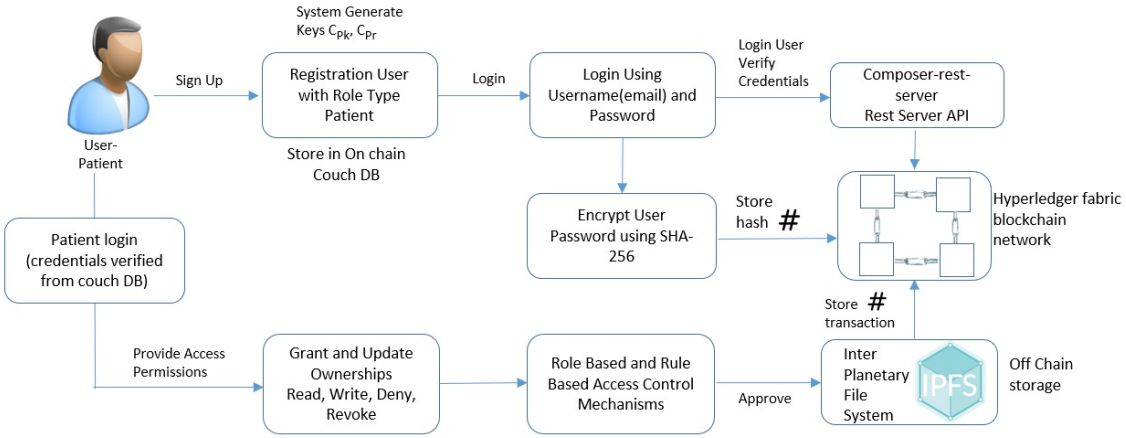


Fig. 4.9 Providing Access Permission.

rules based on role-based and rule-based access control mechanisms for refined and restricted access to medical records. Smart contracts will be executed during user interaction to identify requests, validate requests, update records and grant access permissions to medical records.

4.5.4 Retrieval of Records

Retrieving a medical record can be performed through a series of transactions. The process begins with a patient who uploads his data in IPFS via public key encryption. For the clinician or stakeholder who has access to the record for a particular session from IPFS, the system automatically generates a composite data view P_{Cv} which requires encryption with session key S_k. Additionally the session key will be encrypted with the clinician's public key C_{pk} for secure transfer. The clinician updates the medical record on arrival and encrypts it with the session key before storing it in the IPFS. The system notifies the patient regarding the updates on the medical record that decrypts the updated medical record UP_{Cv_i} with the shared session key. The patient further encrypts the updated record with the patients' public key, commits the updates to the original record and uploads it to IPFS. Moreover, the patient can decrypt his record using his private key from IPFS and upload the encrypted record using the patient's public key. All the transactions which occurred will be hashed by utilizing smart contracts and added to the healthchain. The step-by-step explanation is shown in Fig. 4.10 .

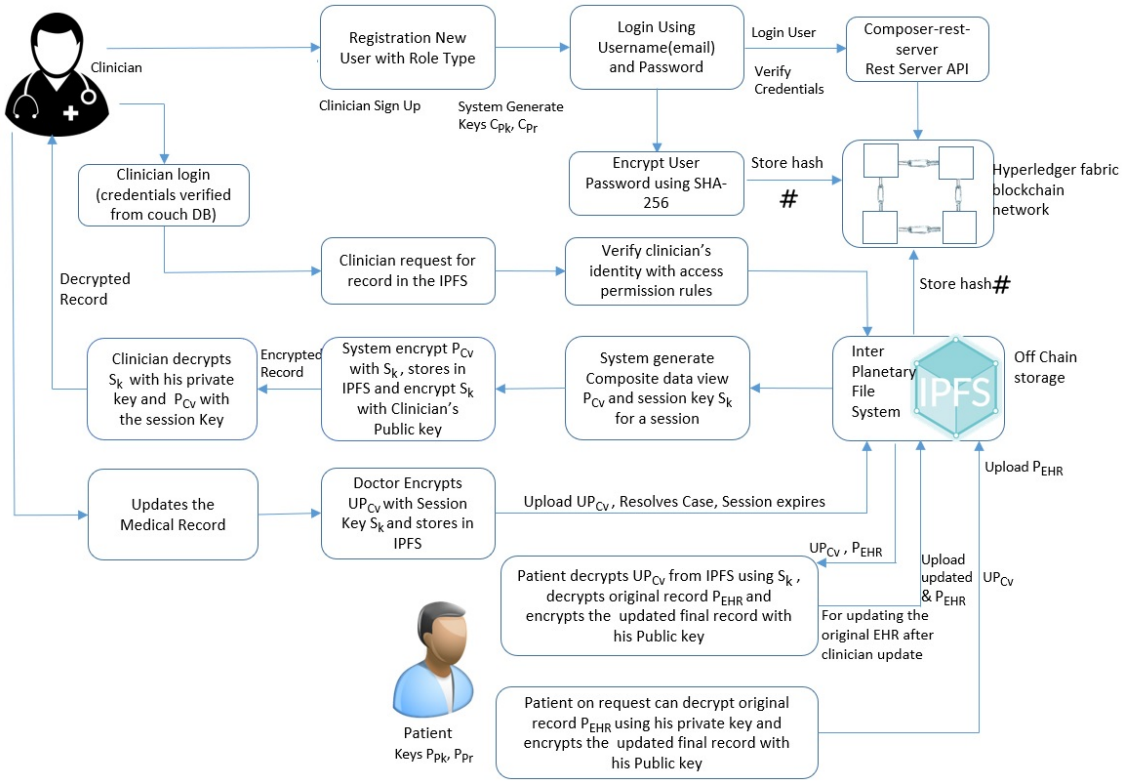


Fig. 4.10 Retrieval of Health Records.

4.6 Prototype Implementation and Results

For the implementation of our proposed Healthchain framework, we initially employed a private Hyperledger Fabric blockchain viz healthchain in a Linux environment. Smart contracts are deployed for every transaction in the Healthchain, the IPFS storage system is utilized and developed network entities are developed to build the Healthchain framework. The following are the main components used for the simulation environment and Table 4.3 presents the machine configurations.

The prototype is a user-centric model to process healthcare records using the blockchain network, ensuring the data ownership of individuals by preserving data security, privacy, data scalability and data integrity. This prototype is designed with a few stakeholders namely doctor (clinician), patient, receptionist and pharmacist that build a private healthchain framework. The framework's flow is detailed as follows:

• Similar to a web application, the URL of the framework is visible to users irrespective of the blockchain technology used at the rear end.

Table 4.3 Development Environment for the Proposed Framework.

Component	**Description**
Operating Systems	Ubuntu Linux 16.04 64 bit
IDE	Hyperledger Composer
CPU	(Intel(R)Core(TM)i5-8500 CPU @ 2.5GHz 2.7GHz
Memory	8 GB
Node	v8.15.0
CLI Tool	Composer REST Server
Docker-compose	Version 18.09.2
Python	v2.7.12
Blockchain Network	Hyperledger Fabric
Framework Tools	Visual studio code
Programming Language	Angular4,Node.js,composer modeling language
On-chain Database	CouchDB
Off-Chain Database	IPFS

• The framework allows the user to signup with vital details like unique id, username, email address and password and the values will be stored in the onchain database, couchDB.

• The user can successfully log in if the username and password matches with the data stored in couch DB by querying the blockchain.

• A doctor who has logged in can upload the medical records to the IPFS by encrypting with the users' public key thereby using public key encryption. The hash value generated by IPFS will be maintained in the couchDB, onchain database of the blockchain and thus preserves data integrity.

• A patient who is logged in will be able to grant and deny accesses such as read, write, and update permissions to the stakeholders on their medical records, thus maintaining restrictive access control.

The illustration of EHR access in Healthchain is presented in Fig. 4.11, Fig. 4.12, Fig. 4.13 and Fig. 4.14. Fig. 4.11 shows REST API that exposes the CouchDB, state database of the blockchain. The data can be queried from the onchain state database via the REST API. Fig. 4.12 (a) is the User Sign Up in which the Patients, Doctors, Chemists and Receptionists can register in the healthchain using their roles.

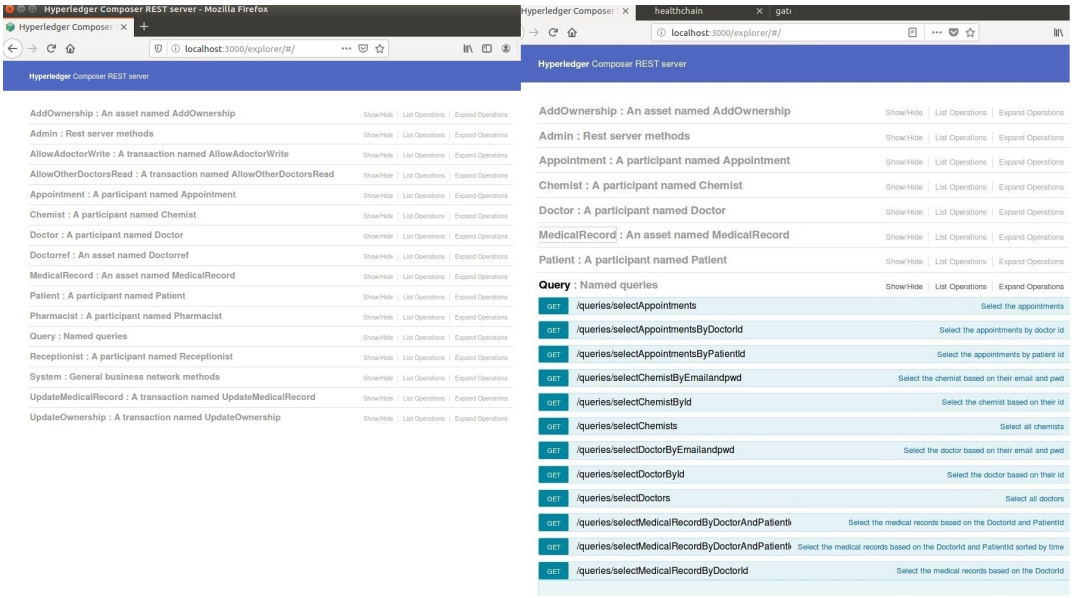


Fig. 4.11 Illustration of EHR Access in Healthchain.

After registration, the user can login with their email address and password by choosing their user type as shown in Fig. 4.12 (b). According to the role type Patient, the patient can view his profile, book an appointment for the doctor, view the medical records and add ownership to the doctor on his medical records as presented in Fig. 4.12 (c).

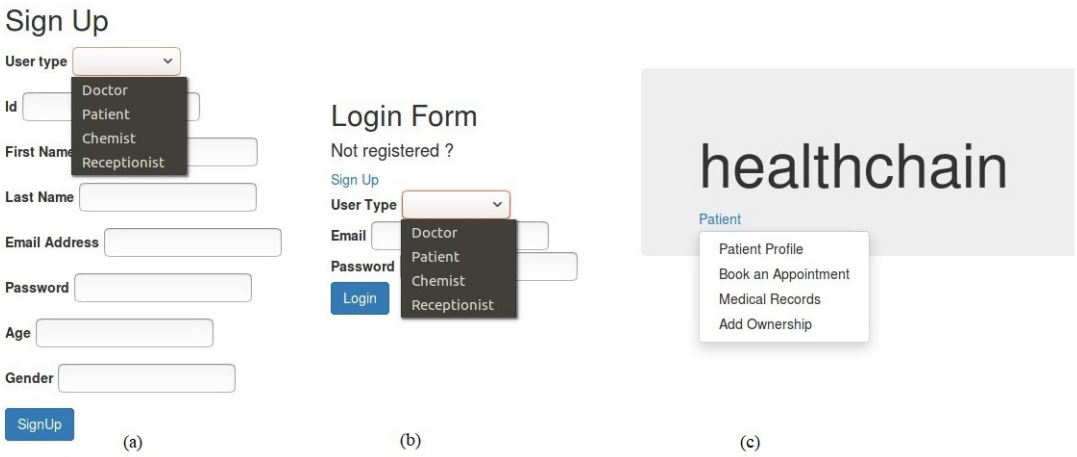


Fig. 4.12 Illustration of EHR Access in Healthchain.

The patient can book his appointment via the Receptionist and the Receptionist can update the participant using patient ID by accepting or rejecting the appointment. After the approval of the appointment by the receptionist, the patient can consult the doctor and

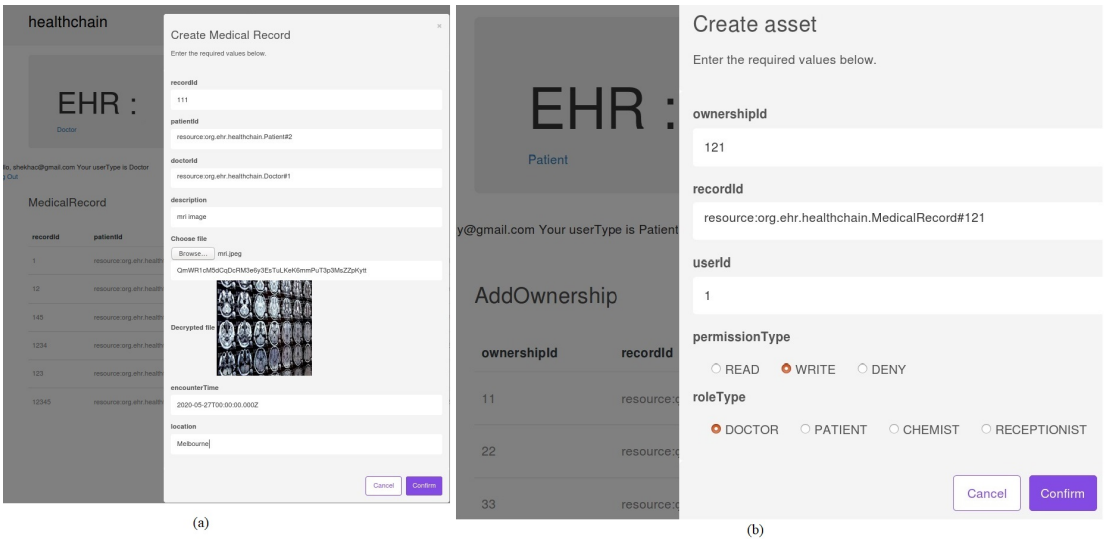
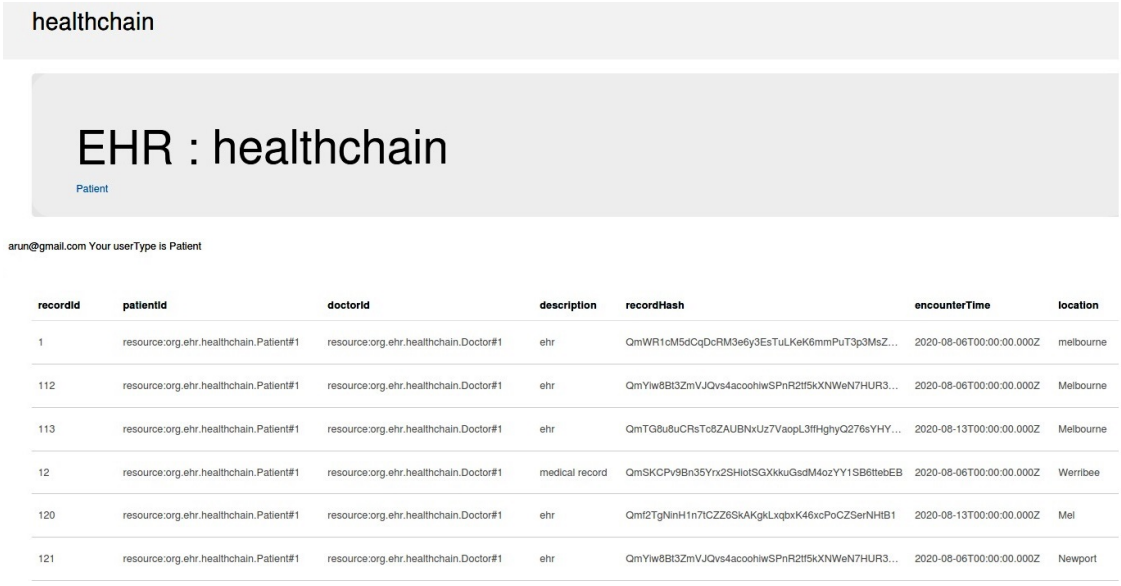


Fig. 4.13 Illustration of EHR Access in Healthchain.

the doctor can create a medical record for the patient. The clinical notes or the diagnosis results can be uploaded to IPFS using public key encryption for a session and IPFS returns the hash of the encrypted record which is stored in the couch DB i.e. blockchain as illustrated in Fig. 4.13 (a). Being a patient-centric blockchain, patients can also provide access permissions such as read, write and in certain situations where the patient wants to revoke access to a doctor on his medical records, permission to the record can be denied as seen in Fig. 4.13 (b). Moreover, the patient can view the medical records added by the doctor as a data provenance [34] shown in Fig. 4.14.



recordId	patientId	doctorId	description	recordHash	encounterTime	location
1	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	QmWR1cM5dCqDcRM3e6y3EsTuLKeK6mmPuT3p3MsZ...	2020-08-06T00:00:00.000Z	melbourne
112	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	QmYiw8Bt3ZmVJQvs4acoohiwSPnR2tf5kXNWeN7HUR3...	2020-08-06T00:00:00.000Z	Melbourne
113	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	QmTG8u8uCRsTc8ZAUBNxUz7VaopL3ffHghyQ276sYHY...	2020-08-13T00:00:00.000Z	Melbourne
12	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	medical record	QmSKCPv9Bn35Yrx2SHiotSGXkkuGsdM4ozYY1SB6ttebEB	2020-08-06T00:00:00.000Z	Werribee
120	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	Qmf2TgNinH1n7tCZZ6SkAKgkLxqbxK46xcPoCZSerNHtB1	2020-08-13T00:00:00.000Z	Mel
121	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	QmYiw8Bt3ZmVJQvs4acoohiwSPnR2tf5kXNWeN7HUR3...	2020-08-06T00:00:00.000Z	Newport

Fig. 4.14 Illustration of Provenance in Healthchain.

4.7 Analysis of the Framework

To validate the functional capability and to evaluate the performance of the prototype, some test cases are explored. Four case studies are investigated to assess the performance of the Healthchain framework systems which are illustrated in terms of efficiency, storage, security and scalability.

• Case I : Efficient storage of Health Records

• Case II : High Degree of Security

• Case III : Enhanced data privacy

• Case IV : Improved data scalability

4.7.1 Efficient Storage of Health Records- Case I

The efficient storage of health records in the Interplanetary file system has been tested against a few cases as listed in Fig 4.15. The first test case verifies if a doctor can upload health records or diagnosed test results on IPFS. The implementation results in Fig 4.13 (a) show that the authenticated doctor can have write access for the medical records and upload encrypted records into IPFS. A public key encryption algorithm is used for

S.No	Test case	Description	Outcome
1	Verify if a doctor can upload medical records on IPFS	The doctor authenticated by the patient can have write access for the record and upload encrypted records into IPFS	Passed
2	Verify if a doctor can view the medical records on permission	The doctor authenticated by the patient can have read access for the record	Passed
3	Verify if a patient can view the medical records	The patient can see the provenance history of their records	Passed
4	Verify if a record is uniquely identified	Each medical record is uniquely associated with a patient id and doctor id	Passed
5	Verify if an encrypted record can be effectively retrieved	The updated medical record can be encrypted with doctor's session key for storing in IPFS and updated record can be decrypted by using patients' session key at the patient side	Passed

Fig. 4.15 Storage of Health records.

encrypting the medical records on to the decentralized storage IPFS. The second case is tested if a doctor has read access permission to the medical records and is successfully verified as the doctor has been authenticated by the patient. Furthermore, it tests that a

patient can view the medical records and Fig. 4.14 portrays the provenance history of the medical records. Moreover, the system is tested against whether a record can be uniquely identified or not and has been successful as every medical record is uniquely related with a doctor ID and patient ID. Additionally, the system has been checked to see if an encrypted record can be effectively retrieved after decryption and has been successful as shown in Fig. 4.13 (a). The outcome is successful as the updated record can be encrypted with the doctor's session key for storing in IPFS and the updated record can be decrypted by using the patients' session key at the patient side.

4.7.2 High Degree of Security- Case II

The degree of security in healthchain has been verified against a few test cases as shown in Fig. 4.16. The first case is tested and successful as the users' password is encrypted before storing user authentication information in the couch database. The second test case verifies the degree of security to check whether the medical records are encrypted on the IPFS and returns a unique hash for the encrypted record as shown in Fig. 4.13 (a). The

S.No	Test case	Description	Outcome
1	Verify if the user's password is encrypted in healthchain	To maintain security, user's password can be encrypted before storing in the state database	Passed
2	Verify if the medical records are encrypted on IPFS	The medical record is encrypted using the patients' public key and uploaded into IPFS and returns a hash value for the encrypted record	Passed
3	Verify if Public Key Infrastructure is used	Two keys public key and private key have been used for user identification	Passed
4	Verify if session is maintained for the user	If the user has not signed out of the application and if session is not expired, the application session will be maintained	Passed
5	Verify if the rest API being used is secured	The state database, couch DB of the healthchain has exposed a rest api that also need to be secured	Can be added in future work

Fig. 4.16 Degree of Security.

outcome is favorable as the medical records are encrypted using the patients' public key before being uploaded into the IPFS. Furthermore, the prototype has also been verified with the usage of the public key infrastructure and was found successful since public and private keys are used for user identification. The prototype has also been tested to check whether the session has been maintained and found successful as long as the user has not

signed out from the application and the session has not expired.

4.7.3 Enhanced Data Privacy- Case III

Heathchain employs several privacy preserving mechanisms. The data privacy in Healthchain is determined based on the permission to access the healthcare records. The access control for the medical records is tested against a few test cases as listed in Fig 4.17 . The initial case is verified and successful as the users can view the homepage

S.No	Test case	Description	Outcome
1	Verify if a user can view the homepage based on user type in healthchain	When the url is exposed to the user, homepage can be seen with user type for an existing user	Passed
2	Verify if a patient can grant or revoke access of the medical record	The patient has all the permissions to grant or revoke access from other user types to maintain privacy	Passed
3	Verify if a patient can provide access permissions	The patient has the permission to provide read, write and deny access permission according to the role type	Passed
4	Verify if a doctor can view the medical record using security token	Since the user can revoke access of their medical record from the doctor, providing access again is troublesome and hence some session token has been added into the framework that expires after the session	Passed

Fig. 4.17 Enhanced Data Privacy : Access Control.

based on their user type as shown in Fig. 4.12 (c). Additionally, the system has been tested to check whether a patient can provide grant or revoke access to the health records to the stakeholders and has been successful in preserving data privacy. Furthermore, the system is also tested to see whether the patient can provide access permissions to the stakeholders. From the simulation results, it can be seen that patients can also provide access permissions such as read, write, and in certain situations where the patient wants to revoke a doctor's access to his medical records, permission to the record can be denied as shown in Fig. 4.13 (c).

4.7.4 Improved Data Scalability- Case IV

Healthchain is well-founded on various notions to promote scalability. This research further contributes to data scalability by storing the hash value of medical records on

chain and encrypted data off chain, in the decentralized storage, IPFS. The scalability of data has been examined against a few test cases as shown in Fig. 4.18. A record

S.No	Test case	Description	Outcome
1	Verify if a huge file can be stored on IPFS	Upload a medical record with size > 100 MB	Passed
2	Verify if a small file can be stored on IPFS	Upload a medical record with size < 10 MB	Passed
3	Verify if the time taken to store and retrieve the medical record is acceptable	The time taken to upload the medical record is few milliseconds	Passed
4	Verify if files or medical records with different extensions can be uploaded in IPFS	Files with different extensions such as video, audio files can be added in a later stage of this research work	Can be added in future work

Fig. 4.18 Improved Data Scalability.

of 100 MB was uploaded at a time to IPFS and has been successful which determined the scalability of the system. Considering the machine configuration, the system also verified that the average time taken by multiple users for the uploading and retrieval of the record was less than 60 seconds. A detailed view is portrayed in Fig. 4.26 and Fig. 4.27. Therefore, it can be concluded that the system is able to handle a large dataset at low latency.

4.7.5 Smart Healthcare and Healthchain

It is extremely vital that healthcare data access is managed with extreme diligence. Healthchain could improve cyber defense capability, as the platform is secured from malicious attacks by its consensus mechanism, immutability, encryption techniques, Authentication, Authorization and Accounting (AAA) capabilities and smart contract functionality. In comparison to a normal database, the combination of hashing and cryptography, as well as its decentralised structure, makes it extremely difficult for any party to tamper with it. The Healthchain implementation is associated with a private blockchain topology that is resilient to external and internal security threats where identifiable blockchain peers and orderers would immediately expose their identity in the event of misbehaviour. All validator identities are revealed in the case of a permissioned private blockchain consortium using Healthchain. As a result, if the actor

(i.e., validator) tries to tamper with EHRs or blockchain transactions, their identity is revealed. In this research, peer nodes use CouchDB as the local state database to store data and couchDB stores hash value of the encrypted data for each transaction. The huge size records will be stored in different nodes in the IPFS after encrypting using special public key encryption mechanism. A detailed algorithm and the cryptographic process is explained in Algorithm 1 and Fig. 4.4.

We propose a special cryptographic encryption scheme that guarantees to simultaneously achieve authenticity, confidentiality, unforgeability, and access control. For every record access, the proposed Healthchain uses this encryption technique that encrypts the composite view of the data with the generated session key and encrypts the session key with the stakeholder's public key to enhance the security. Besides, this work also achieves access control, authenticity, interoperability and node scalability. Healthcare organisations and individuals can only access the data for a specific session by adopting appropriate security measures based on special encryption mechanism and identity management using access control, consequently addressing requirements such as availability, security, and privacy. HLF can also mitigate malicious attempts with its built in x.509 Certificate authority and also make use of SHA-256 algorithm to prevent spoofing and tampering of data. Being a patient centric system, patient as the owner of the EHR has the only power to provide various access rights and permissions for data sharing and accessing sensitive information to various stakeholders in the Healthchain ecosystem. Because this approach stores data in a distributed manner, single-point-of-failure is also reduced. The performance evaluation demonstrated that system is foolproof, unforgeable and secure in storing patient data, never permits tampering, and shares the data only with the patient's agreement. Internal attacks can be eliminated in this framework as it requires to reveal user identity to participate in the Healthchain network. In addition, access will be provided to the attribute dataset only to authorised users for a particular session without disclosing the entire patient record. The external malicious attack can be ruled out with the hash being stored in the blockchain that maintains secrecy and data privacy. Healthchain being a patient centric system, can prevent intentional and unintentional attacks as all the access permissions require the consent of the patient to perform any transactions in the network. Therefore attacks

caused by both intentional and unintentional access can be mitigated.

4.7.6 Comparative Analysis of the Proposed Framework with Existing Blockchain Techniques

This section performs a comparative analysis of the proposed framework with the existing blockchain techniques in terms of major privacy preserving requirements viz data integrity, data privacy, data security, confidentiality and scalability. The proposed framework is compared against the existing blockchain based implementations such as [129], [67, 146] and [49]. From Table 4.4, it is evident that the proposed system satisfies the shortcomings of the existing systems in terms of data security, privacy and scalability. This section also describes how the proposed framework satisfies the privacy preserving requirements.

Table 4.4 Comparative Analysis.

Scheme	Data Integrity	Data Privacy	Data Security	Confidentiality	Scalability
MedChain[129]	✓	✗	✓	✓	✗
Wang & Song[146]	✓	✗	✓	✓	✗
Blochie[67]	✓	✓	✗	✓	✗
Blockchain for IoT[49]	✓	✓	✓	✓	✗
Proposed Framework (Healthchain)	✓	✓	✓	✓	✓

Data Integrity : Data is immutable and tamperproof as the data is stored as hash values in each block and each block stores the hash value of the previous block in this blockchain framework. The trust in this blockchain framework is based on the consensus, digital signature and the designed cryptographic algorithm despite relying on a third-party provider. Since all the blocks are linked, any alteration in the original data will result in a change in its hash value and it is computationally difficult to tamper with the ledger, such that the non-tampering of the medical record is also explicitly guaranteed. In addition, the original data is stored in IPFS after performing a special cryptographic encryption technique and IPFS stores the data in multiple nodes if the size of the data is higher than a defined threshold.

Data Privacy : This framework is of paramount significance to health record data privacy and patient privacy. In addition to special encryption mechanisms that ensure data security, access control permission rules have been implemented in the system to safeguard the data privacy of patient health records. The framework ensures fine-grained access control by integrating role, rule and attribute-based access control permission rules for any data request. Secondly, an unauthenticated data requester cannot access data location since the blockchain only stores the hash value of the encrypted medical record. Thirdly, if the data requester attributes do not meet the access policy embedded in the network archive file, it is also impossible to acquire any real medical record data from the blockchain public information.

Data Security : Data Security is a crucial feature as the EHR is cryptographically stored and dealt with in the proposed system. This blockchain framework stores only the hash of the encrypted data on chain and the actual huge data is stored after encryption in the offchain storage. Since the framework is a patient-centric approach that provides authenticated access permissioned by the patient which guarantees the security of the health records. Also, the smart contracts' functionality combined with blockchain solutions embrace high-level encryption and ensures patient confidentiality in their health care information. In addition, the data stored on IPFS is encrypted using a special cryptographic algorithm to establish robust blockchain data solutions.

Confidentiality : In this framework, every health record of the patient will be stored in the IPFS after encrypting with the patients' public key and allows only the permissioned or authenticated users to access the record for a particular session. Since the framework is a patient-centric approach in which the patient has complete control, unless for emergency situations to provide access permissions to the stakeholders, the confidential nature of health data is preserved.

Scalability : The proposed scheme preserves most of the privacy requirements and provides cryptographic storage of health records in IPFS and thereby resolves the

scalability issue in the existing techniques. The scalability of the proposed system proves that the system is capable of processing large datasets at low latency as shown in Fig. 4.26 and Fig. 4.27.

4.7.7 Performance Analysis and Discussion

The evaluation matrix for the framework which specifies the stakeholders, functions and solved problems to facilitate privacy preservation requirements is shown in Table 4.5. The framework is user-oriented and handles the efficient storage and transfer of medical records ensuring the data ownership of the individuals, patient confidentiality and data integrity. By adopting access control mechanisms, clients can manage their own private information without jeopardizing confidentiality. Meanwhile, each requisition and update from the stakeholders viz receptionist, doctor, patient and pharmacist are reflected in the couchDB, state database of the healthchain. The patients can handle the access control mechanism by granting or revoking access to the medical records to the stakeholders thereby maintaining user and data privacy. Data security and patient confidentiality is attained by data storage using public key encryption that secures the user data.

Several experiments have been carried out to analyse and evaluate the performance of the proposed blockchain system. The assets defined here are: (a) Medical Record (b) Referrals (c) Prescription (d) Add Ownership. The transactions are: (a) Create Medical Records (b) Update Medical Records (c) Allow Doctors Write (d) Update Ownership (e) eReferrals to other Doctor (f) ePrescription to Pharmacist. Health care data encompasses a wide range of data collection processes, both public and private, such as health surveys, administrative and billing records, and medical records used by various agencies such as hospitals, physicians, and health plans. A test dataset containing images and text data of varying sizes is used in this analysis. The first experiment calculated the transaction latency of the proposed blockchain framework as shown in Fig 4.19. Transaction latency is the amount of time taken for the transaction to commit and is available across the network nodes. If there are n number of nodes in the blockchain network, T_{L_n} is the

Table 4.5 Evaluation Matrix.

Stakeholders	Functions	Solved Problems
Patients	Provide access control, User login, Encrypted data storage, Decentralisation, Data provenance, Data retrieval	Data confidentiality, Authentication, Privacy, Data Scalability, Authorization, Non-repudiation, Data integrity
Doctors	User login, Data storage with encryption, Secure e-Referral, Data retrieval, Prescription management	Authentication, Confidentiality, Scalability, Non- repudiation, Integrity, Security
Pharmacists	User Login, Prescription Management	Authentication, Non- repudiation, Integrity
Receptionists	User login, Appointments management	Authentication, Confidentiality

transaction latency, T_{C_n} is the confirmation time in the network nodes and T_{S_n} is the transaction submit time in seconds then;

$$T_{L_n} = T_{C_n} - T_{S_n} \tag{4.5}$$

Seven sets of writing transactions to the network ledger were performed in various transaction sets within a range of 5,10,15,20,30,40 and 50 as shown in Fig 4.19. Considering the machine configuration in Table 4.3, it is clear that the initial set of 5 transactions took an average of 80 seconds to commit across the network and the final set of 50 transactions took an average of 160 seconds. The experimental result is further extended to the Montecarlo Simulation environment for determining the transaction time for the number of transactions in the range of 50 to 300. It can be seen that an

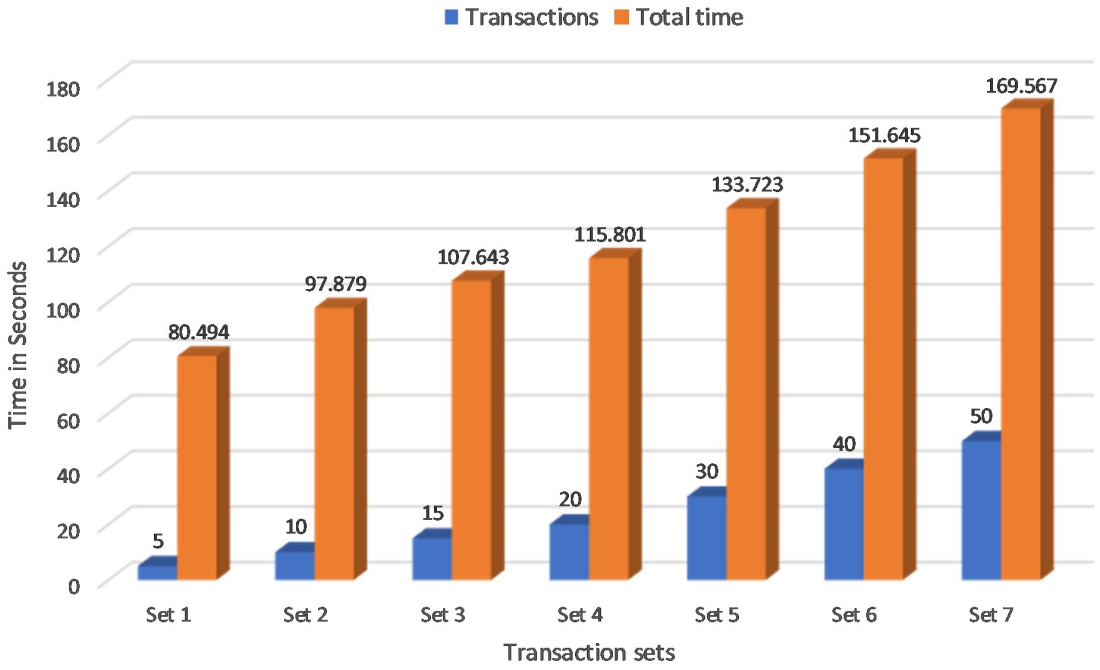


Fig. 4.19 Transaction Latency.

average of 450 seconds was required to commit 300 transactions in three peer nodes as shown in Fig 4.20. Therefore, it is evident that the time taken to execute transactions

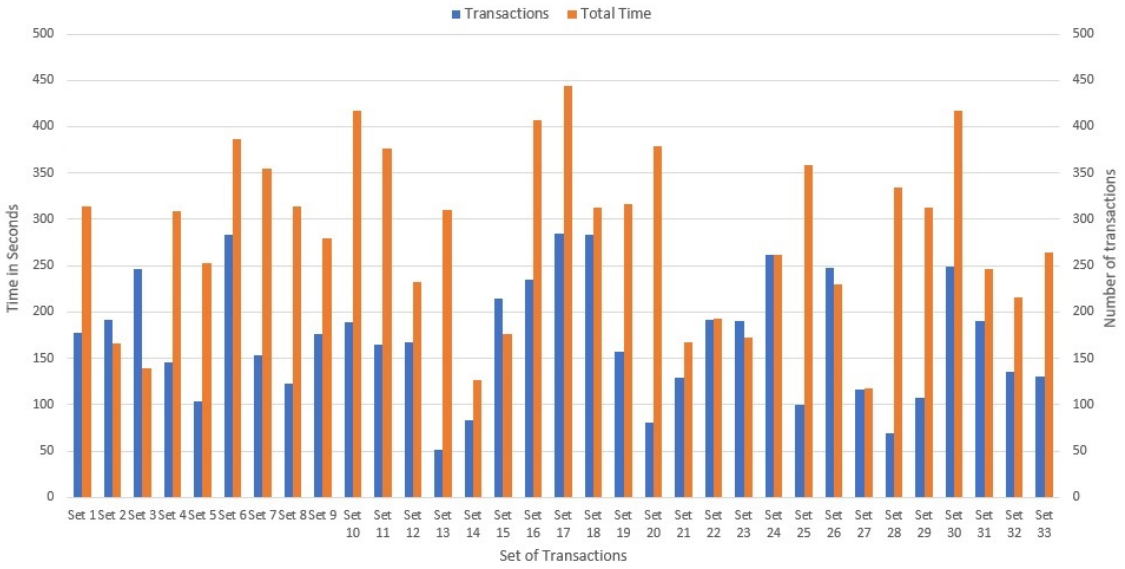


Fig. 4.20 Transaction Latency: Montecarlo Simulation.

increases with an increase in peers and an increase in the number of transactions. Fig 4.21 shows a comparative analysis of transaction latency of 1 Org 1Peer, 1 Org 2Peer and 1 Org 3Peer. For seven sets of transactions ranging from 5, 10, 15, 20, 30, 40

and 50, it is clear that for 5 transactions, 1 Org 3Peer takes 80 secs to commit across the network in which 1 Org 2Peer took 67 secs to commit and 1 Org 1Peer took an average of 45 seconds to commit across the network. Therefore, it shows that more peers and a higher number of organisations exhibit higher latency. The second experiment

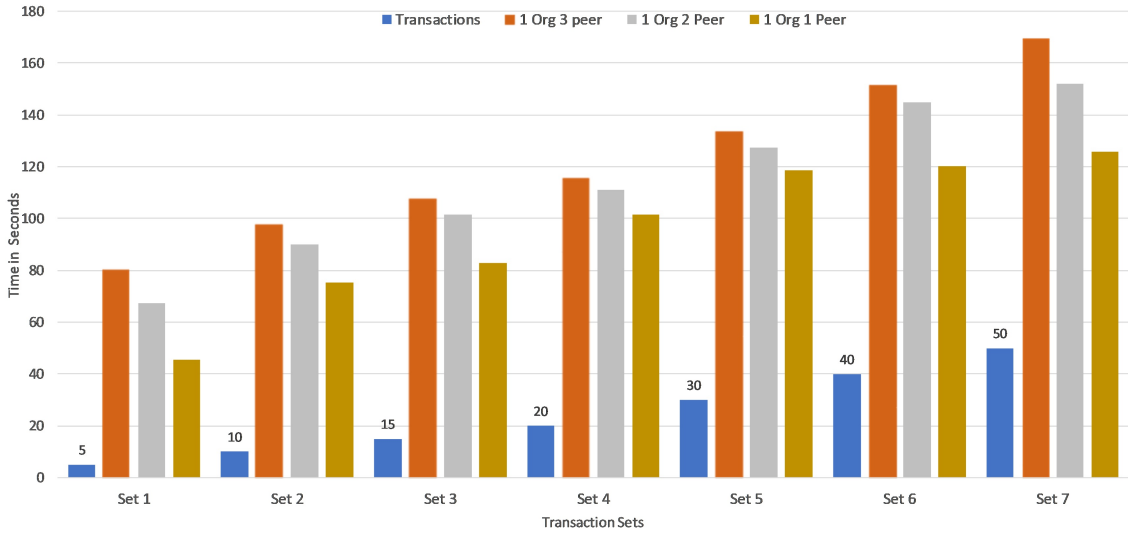


Fig. 4.21 Transaction Latency: Comparative Analysis.

calculated the transaction throughput of the proposed blockchain framework. The transaction throughput is the rate at which the blockchain System Under Test (SUT) commits valid transactions in a defined time period at all network nodes. If there are n number of nodes in the blockchain network, T_{T_n} is the transaction throughput, T_{ct_n} is the total number of committed valid transactions in the network nodes and T_{tot} is the total time in seconds then;

$$T_{T_n} = T_{ct_n}/T_{tot} \tag{4.6}$$

Fig 4.22 portrays the transactions per minute (TPM) for various sets of transactions. This experiment runs 7 sets of transactions to determine the TPM in the proposed system. The first set has 5 transactions and took approximately 80 seconds to commit in the network. As a result, the rate of valid transactions across the SUT is 4 TPM in the network. Similarly, the last set of 50 transactions took approximately 160 seconds to be available across the network and thereby can commit 18 TPM. The x-axis indicates the transaction set, y-axis as time in seconds and secondary y-axis for TPM. Fig 4.23 shows a comparative analysis of transaction throughput that calculates the TPM of the proposed

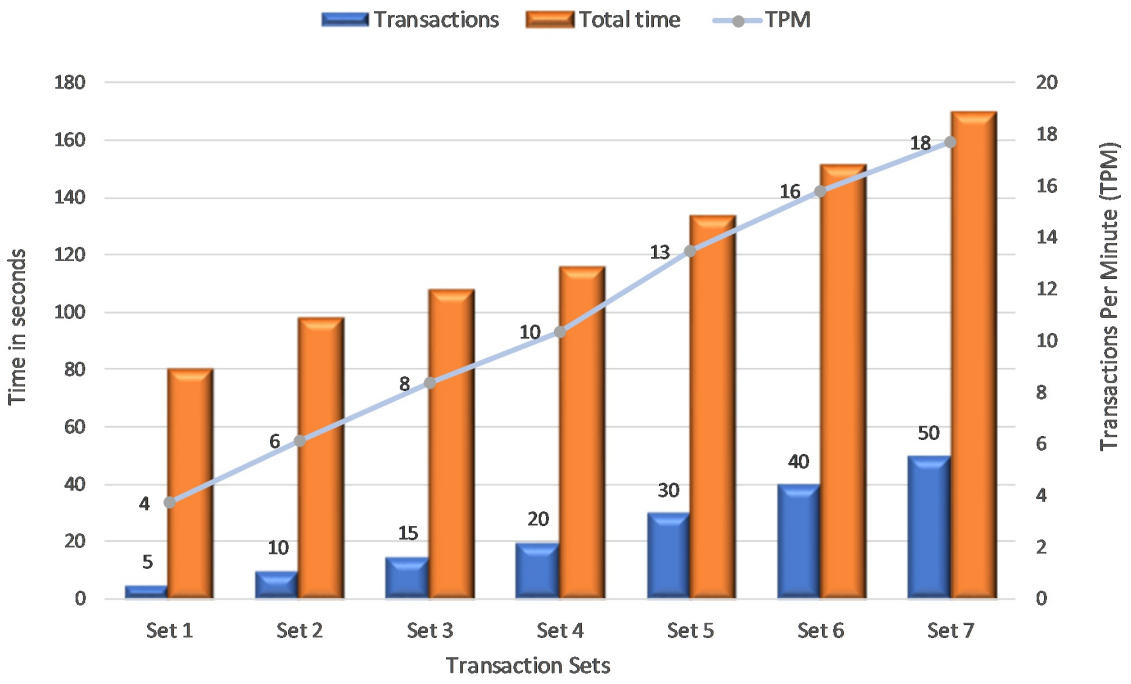


Fig. 4.22 Transaction Throughput.

framework for 1 Org 1Peer, 1 Org 2Peer and 1 Org 3Peer. From Fig 4.23, it is evident that based on the transaction latency in Fig 4.21, the rate of valid transactions across the SUT is slightly higher for 1 Org 1Peer compared to 1 Org 2Peer and 1 Org 3Peer. The

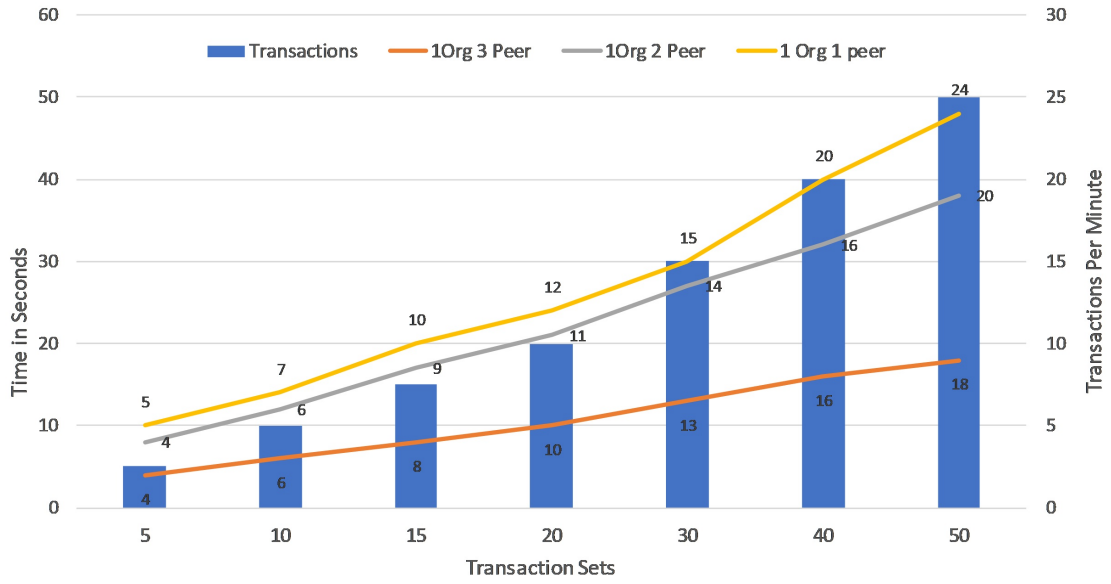


Fig. 4.23 Transaction Throughput: Comparative Analysis.

asset latency is the time taken by the SUT to successfully load and write the assets to the

couchDB. If there are n number of nodes in the blockchain network, A_{L_n} is the Asset Latency, T_{Res_n} is the Response time and T_{Sub_n} is the asset submit time in milliseconds then;

$$A_{L_n} = T_{Res_n} - T_{Sub_n} \tag{4.7}$$

Fig 4.24 shows the varying assets size in bytes of five concurrent users in the proposed system and it is obvious that it took an average latency of 2.7 seconds to commit asset write updates in the couchDB across the network. It is observed that the asset size of 154K bytes took an average of 2.6 seconds and 15478K byte size took an average of 2.7 seconds to commit write updates in the CouchDB. We also extended the experiment to project the number of concurrent users in a range of 5 to 100 and byte size in a range of 154K bytes to 20574K bytes to determine the variation in asset latency through Montecarlo simulation and it took an average latency of 3.0 seconds to commit the asset updates in the ledger as shown in Fig 4.25. Considering the machine configuration in Table 4.3, system efficiency is higher as it is obvious that even if the number of users increases from 5 to 100 and asset size increases in the SUT, it required a marginally small increase in time to commit the asset updates to the couchDB across the network.

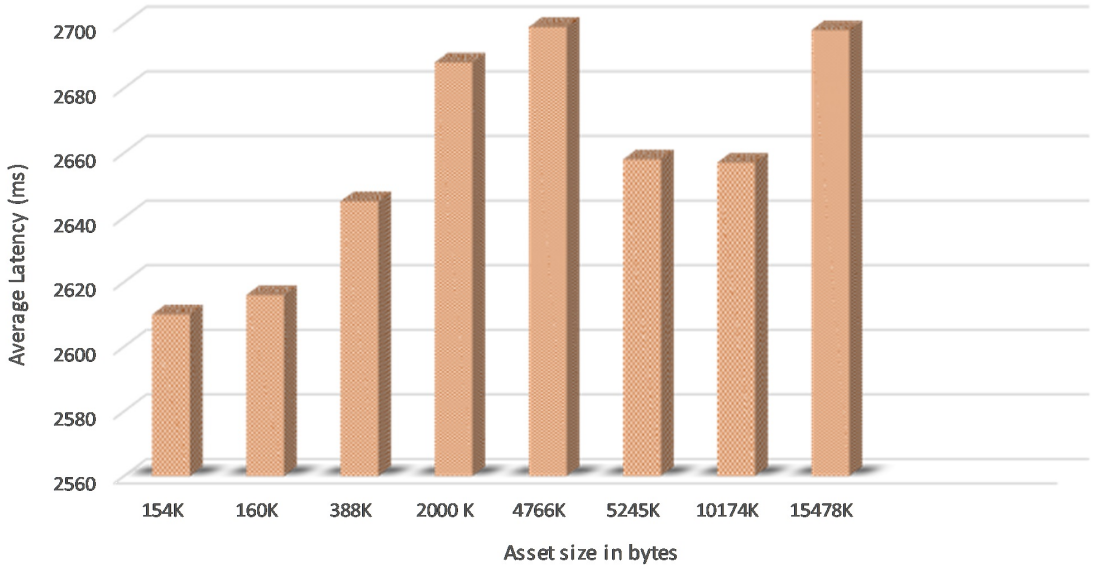


Fig. 4.24 Asset Latency.

Scalability and efficiency have been achieved by uploading a record of 150 MB at a time to the IPFS and the average time taken for five concurrent users uploading and

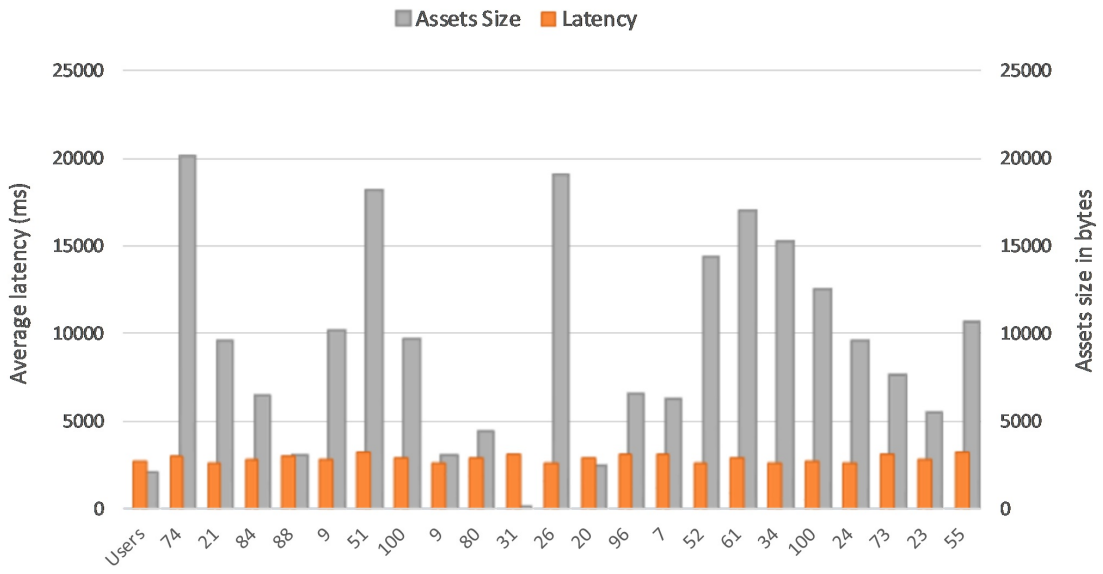


Fig. 4.25 Asset Latency: Montecarlo Simulation.

retrieval of the record was 60 seconds. Thereby, it can be concluded that the proposed system is capable of processing a large dataset at low latency. Data provenance can also be attained via preserving user history in the blockchain thereby safeguarding non-repudiation. Smart contracts combined with blockchain solutions embrace high-level encryption that allows providers, users, patients and clinicians to ensure patient confidentiality in their health care information and ensure it is attack-proof. Furthermore, Healthchain is designed to enhance the scalability of healthcare data by storing hashes on chain and real data in the off chain IPFS. Fig 4.26 and Fig 4.27 demonstrates the scalability of IPFS using both the image data and document data with a size comparison up to 100 MB in size. The results are obtained from the transaction execution of five users concurrently upload and download the data in IPFS. Considering the machine requirements, for a 100 MB image file, the system takes an average time of 65 sec to upload the data to IPFS and download it in an average time of 80 seconds as shown in Fig 4.26. Also, the system takes an average of 65 seconds for uploading time and an average time of 105 seconds for downloading a 100 MB document file as portrayed in Fig 4.27. Healthchain is a patient-driven interoperability framework and employs several security and privacy preserving mechanisms that sustain cyber attacks and internal attacks, however there are still some improvements that could be made to

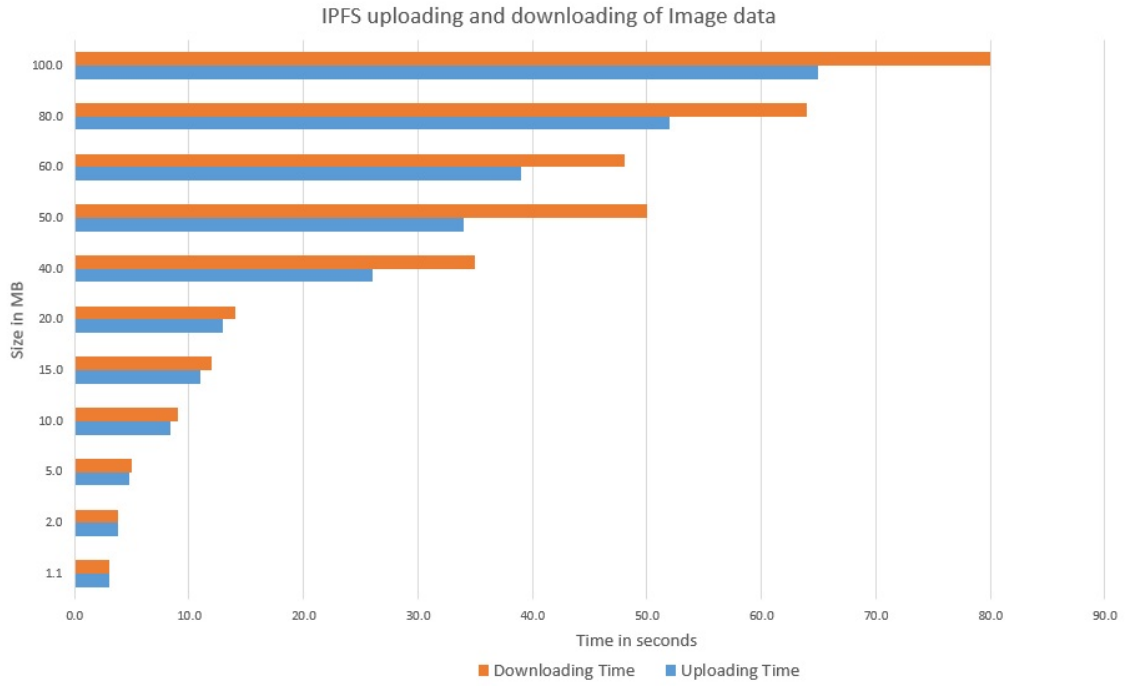


Fig. 4.26 Uploading and Downloading Time Comparison of Image Data in IPFS.

make it a foolproof solution. Initially, the REST API can be made secure via using HTTPS by encrypting communications between client and server instead of HTTP that is being used nowadays. Secondly, we can employ smart contracts on a large scale that

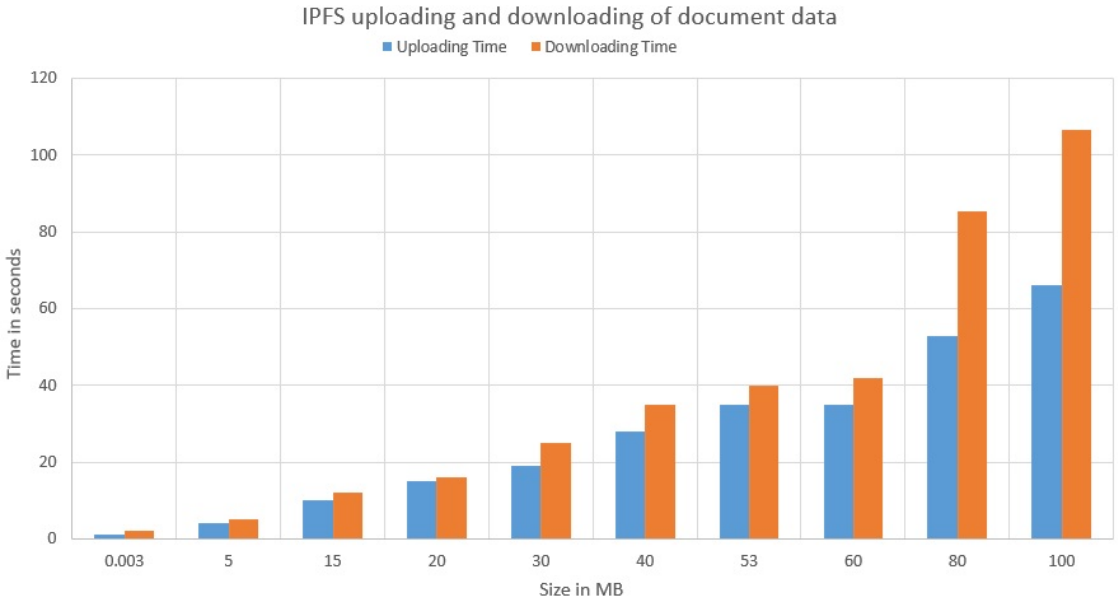


Fig. 4.27 Uploading and Downloading Time Comparison of Document Data in IPFS.

will be executed on a greater number of nodes for the privacy and safety of patients' information to make it tamper resistant. This work can be extended to multiple nodes to

prove the effectiveness of distributed ledger technology in health records as future work. The implementation of different smart contracts on every node and submitting the node to the system requires several stages of verification which is considered future work to prove the efficiency of the proposed system.

4.8 Summary

In this chapter, a permissioned blockchain framework was implemented for secure data storage and to access electronic health records utilizing Hyperledger Fabric and Hyperledger Composer. The main contributions are: (1) since the blockchain is tamper resistant, the system is tamper-proof to handle healthcare records that preserves data privacy, security and integrity; (2) no incentive mechanisms for blockchain mining are included that demonstrate the patients' ownership towards their healthcare data; (3) this research proposes an architecture for securing data storage and providing efficient access control between stakeholders viz patients, doctors, pharmacists and other participants via encryption techniques and access control mechanisms; (4) a working prototype based on Hyperldger Fabric and Interplanetary File System is made to illustrate the system's viability. The proposed methodology has been implemented and evaluated with some use cases for EHRs and consequently, the framework is successful as a reliable health data network.

The result of prototype implementation and analysis proves that the approach is a tamper-resistant mechanism as information will be stored as hash values for every healthcare transaction in the blockchain. Moreover, it has enormous potential to ensure the privacy, security, integrity, confidentiality and scalability of the e-health information. The performance evaluation of the proposed system is complete using empirical research for various scenarios by configuring asset size, block size, various nodes, asset creation time, transaction sets, for evaluation metrics such as transaction latency, transaction throughput, asset latency and data scalability for analysis. Furthermore, this research also explores the technology framework and business processes for blockchain applications. The introduction of this technological innovation which incorporates

cryptographic elements offers a more secure and effective framework to store, transfer and access EHR in the cloud environment efficiently.

Chapter 5

A Healthchain based Smart Contracts System for eReferral in Healthcare using Hyperledger Fabric and InterPlanetary File System

Blockchain is evolving and advancing as a secure and reliable platform for effective and secure data sharing in many areas such as supply chain management, the financial industry, the energy sector, the Internet of Things and most importantly, in healthcare specific implementations. The privacy of EHRs is a major issue while outsourcing data in the cloud or sharing records among stakeholders which includes the leakage of private and sensitive information to unauthorized entities. The standard referral management process includes various discrete steps in the communication between stakeholders through faxed papers, email and telephone that leads to the likelihood of errors, discrepancies and missed connections. In consideration of the identified problem, this chapter aims to: *(i) introduce an efficient referral mechanism employing advanced smart contracts for the effective sharing of healthcare records between clinicians in the healthcare industry; (ii) Hyperledger Fabric as the permissioned blockchain utilising Hyperledger Composer as the rest server which visualizes the couchDB and the*

Interplanetary File System as the decentralised data storage for efficient and secure big data sharing in the healthcare sector; (iii) conduct simulation studies to prove the scalability of IPFS as a decentralised file system and also introduces an efficient encryption technique for the secure storage and transfer of medical records. This referral system is built on a patient-centric model and is limited to authorized providers in the health data network. Through the results, the proposed model demonstrates that healthcare records are not traceable to unauthorised access as the working model only stores the encrypted hash of health records which is effective in data security, enhances data privacy, enhances data scalability, improves interoperability and data integrity when sharing and accessing medical records across the HealthChain network. Therefore, this referral system offers flexibility, scalability and can establish trust among patients, clinicians and other stakeholders.

5.1 Introduction

The previous chapter proposed a permissioned blockchain working prototype based on Hyperledger Fabric as the underlying blockchain technology and IPFS as the decentralised file system for a reliable health data network. In this chapter, the research contributes a Distributed Ledger Technology Smart contract system for efficient eReferral between multiple clinicians in the health data network in the medical industry. Medical referral is the transition of a patient's treatment upon request from one doctor to another. The standard referral management process includes various steps in communication through faxed papers, email and telephone calls. Distributed ledger technology(DLT) also known as blockchain, provides an ideal way to automate the referral process as it provides a secure, real-time data exchange between disparate entities, reducing the likelihood of errors, discrepancies and missed connections. Blockchain technology is one of the cutting-edge solutions that has revolutionized the healthcare industry by facilitating the secure and efficient sharing of health records among stakeholders. Blockchain properties such as immutability, interoperability, shared storage and distributed ledgers in the development of decentralized frameworks

is very promising nowadays. Blockchain is a decentralised, append-only, immutable digital ledger with a chain of blocks which are cryptographically secured by public key encryption standards that ensures immutability, anonymity and block resistance. In addition, an integral mechanism fueling a blockchain network is the consensus protocol which generates, updates and validates each transaction. It also uses scripting technology called smart contracts that comprise the application logic of the system. Being immutable, trustless, decentralised and distributed, blockchain technology offers wide opportunities for combating fraud, reducing operational costs, optimising processes, eliminating duplication of work and improving transparency in the health care industry.

This framework designs eReferrals so that they can be sent and received directly between healthcare providers via secure messaging by employing the smart contract functionality in HealthChain. The theft of EHRs is becoming increasingly pervasive while sharing data due to the poor security and policy enforcement mechanism in the current system. As health records are kept in centralised silo repositories, health data becomes an extremely tempting target for attackers, ranging from Ransomware attacks to the recent malware attack [105]. Most importantly, centralization increases the security risk footprint and requires trust in a single authority. In most countries around the world, centralised health databases are a legal requirement and necessity and therefore require an additional layer of technology to improve their portability and safety. Blockchain data management applications create utility for patients, doctors and healthcare institutes in the areas of patient record access and control, claims and payments management, medical IoT security management and data verification research, and exchange for financial audit and transparency. This work creates smart contracts for various medical workflows, and then data access permissions are managed by the patient in the healthcare ecosystem.

This work introduces the efficient use of smart contracts to ensure a secure exchange of information between providers that encourages physicians to refer their patients to different health care facilities to minimize unnecessary visits to hospitals. This process entails multiple steps which require communication from provider to provider and

patient to provider. A well-structured and effective medical referral system may enhance comprehensive health care for all patients by giving priority to those who need it, reducing health inequalities and limiting the financial burden of unnecessary hospital visits along with the financial burden of health services. This system enables secure and efficient communication with other healthcare providers in the clinical system and strengthens relationships with referrers and promotes quality patient referrals in real time. This chapter also introduces a smart healthcare contract system for managing medical data and streamlining challenging medical procedures. This research builds a patient-centric permissioned blockchain namely Healthchain built on Hyperledger Fabric by utilizing Hyperledger Composer as the rest server API. To avoid failure in third-party servers, this work also presents a secure and efficient decentralised platform viz Interplanetary File System for secure data storage. Moreover, the data at rest is encrypted by an efficient algorithm based on public key encryption standards. This thesis also demonstrates the future use of blockchain in healthcare and the challenges and possible directions of blockchain technology. Fig.5.1 further outlines Healthchains's ability to provide efficient referral between multiple clinicians in the health data network.

- The main contribution of this research is to provide a Distributed Ledger Technology

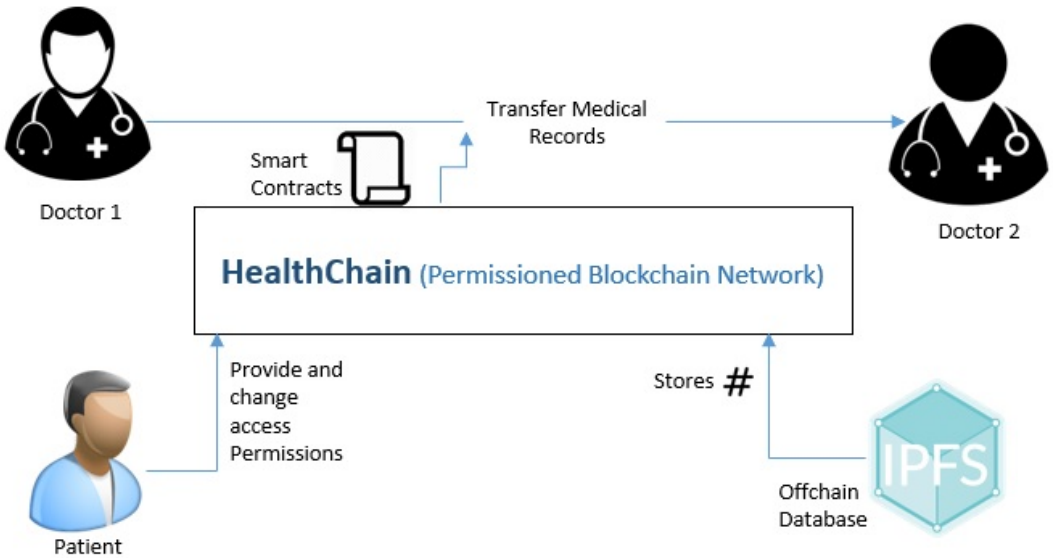


Fig. 5.1 Referral Mechanism between Clinicians.

Smart contract system for efficient eReferral in the medical industry. Fig. 5.1 outlines Healthchains's ability to provide efficient referral between multiple clinicians in the health data network. Smart contracts allow secure and efficient interaction between stakeholders. This work creates smart contracts for various medical workflows, and then data access permissions are managed by the patient in the healthcare ecosystem.

• This research builds a patient-centered Healthchain framework in which patients will have complete control over their medical records maintaining e-health data security, privacy, scalability, and integrity. The Healthchain framework is based on Hyperledger Fabric, a permissioned distributed ledger solutions using Hyperledger Composer and stores encrypted EHRs in the InterPlanetary File System (IPFS) to build this private health chain network.

• Finally, this research addresses the scalability of the healthrecords by storing the hash of health records on the chain to maintain the overall efficiency of the blockchain, and the actual huge data are stored off the chain in a storage framework in IPFS, the decentralised storage. Moreover, the data at rest is encrypted by an efficient algorithm based on public key encryption standards.

Fig.5.1 provides an overview of the referral workflow in the Healthchain framework. Initially doctor 1 chooses to refer a patient to a specialist doctor 2 by initiating the referral after obtaining the patients' approval. The smart contracts written on the back end are invoked for the transfer from one doctor to the next. Doctor 1 decides the amount of information to be shared and the referred doctor will be able to view the referred patients' information such as lab tests and imaging results accordingly. After the specialist diagnosis, another transfer of information including the findings and recommendations of doctor 2 will be uploaded to IPFS after secure encryption. In addition, the referral is only given for a specific session and expires upon completion of the task so the referred doctor loses access to the patient's information and all the transactions will be added to the Healthchain.

5.1.1 Need for DLT Smart Contracts in eReferral

1. Lack of communication, missed or non-returned calls and faxes, lack of coordination among procedures, inadequate patient data are the challenges in today's referral procedure.

2. By driving unprecedented transparency and highly secure data sharing between disparate entities, DLT smart contracts are ideally placed to mitigate communication-based referral problems.

3. Considering DLT as an immutable log of referral transactions, every participant in the referral has an exact copy of this ledger, providing a single source of reality that is decentralised and all participants are aware of any changes to it. Therefore, physicians are no longer left to worry whether a patient received the required treatment, prescriptions or tests.

4. DLT smart contracts are the ideal technology for automating referral management due to their ability to facilitate fast transactions, eliminate data leakage and ensure safe data exchange while maintaining a decentralised single version of the truth.

5.1.2 Transaction Workflow of Hyperledger Fabric

Fig.5.2 shows the transaction workflow in Hyperledger Fabric. A peer is a node that runs on the binary Hyperledger Fabric and each organisation should have peers for hosting ledgers and smart contracts. Each network has its own data stored on peers in a separate ledger and each channel has one or more smart contracts. Applications associate peers to query (get) or invoke (put) information in ledgers as shown in Fig. 5.2. In Fig. 5.2, A denotes Application, P1 is the peer node, smart contract is S1, Ledger is L1 and Orderer node is O1. Whenever a transaction is executed, application A connects with peer node P1 and invokes the appropriate smart contract S1 to update the ledger L1. To generate a query result or a ledger update as a response, peer node P1 then invokes chaincode S1. After receiving the proposal responses, application A builds a transaction from all the responses and sends it to Orderer node O1 for ordering. Orderer O1 performs the

ordering and collects all the transactions from the network into blocks, and distributes the transaction updates to all peer nodes, including P1. Peer node P1 validates all the incoming transactions before updating it to Ledger L1. Once the ledger L1 has been modified, P1 creates an event provided by A to indicate completion.

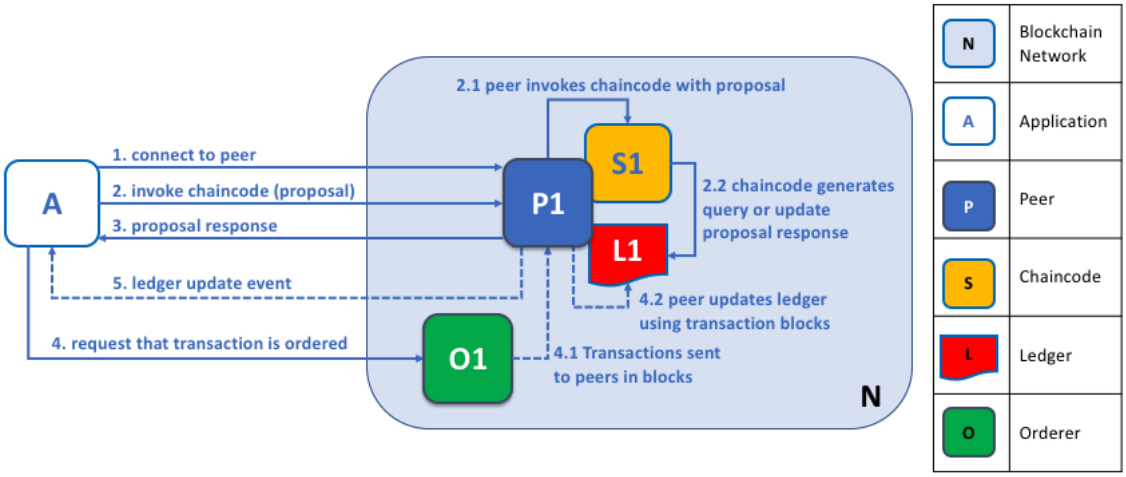


Fig. 5.2 Transaction workflow of Hyperledger Fabric [150].

5.2 Comparative Study of Existing Techniques with the Proposed Work

This section describes the related works on e-health systems using blockchain technology. Bitcoin and Ethereum blockchain were the initial permissionless blockchain implementations that do not have restrictions on their network which means anyone can participate and become a node in the network [44] [108]. The first scheme using blockchain in the healthcare sector that mentioned a Healthcare Data Gateway, the possibility of data sharing on a private blockchain that allows patients to manage their health data without any breach of privacy or security [160]. However, scalability is also a major problem as e-health data is growing rapidly due to data storage on the chain which further leads to blockchain centralization. MedRec is the first permissionless working prototype in healthcare using the Ethereum smart contract functionality for the intelligent representation of medical records stored in individual nodes in the network. Though there is no single point of failure, it fails to address scalabiity issues [11].

Table 5.1 Comparative Study of Existing Techniques with the Proposed Work.

Reference	Security	Privacy	Integrity	Mining	Scalability
Yue [10]	✓	✓	✗	✗	✗
MedRec [1]	✓	✓	✓	✓	✗
Dub [5]	✓	✓	✗	✗	✗
Li [6]	✓	✓	✗	✗	✗
Morgan [8]	✓	✓	✗	✗	✗
Chen [2]	✓	✓	✗	✓	✗
Base paper	✓	✓	✓	✗	✓

Ancile is another permissionless framework that stores the hash references on the chain and employs proxy re-encryption for the secure transfer of medical records but fails to efficiently store health records [43]. Ancile and Medrec have issues with scalability, which can be overcome by using IPFS via the secure storage offchain rather than the chains itself. Dubovitskaya presented a secure data sharing blockchain based on oncology that utilises a local database and cloud infrastructure for the storage of encrypted patient data [48]. A novel patient-centered architecture has been proposed for fine-grained and flexible data access control using ABE to encrypt EHR data [86]. A permissioned blockchain implementation called QuorumChain, allows only a few users or nodes to vote on which data or blocks are to be added to the chain through a smart contract which reduces the complexity of the voting process [106]. Another framework that utilizes Hyperledger Fabric as the blockchain mechanism for healthcare data sharing employs mining incentives for providers to access records and also involves a certification authority that oversees every healthcare service [31]. Many studies have shown that introducing electronic medical referral systems will enhance the referral process, accessibility and coordination between family doctors and specialists, thus increasing patient and medical satisfaction [54] [145]. Compared to the current model of sharing health information, patients chose to use blockchain-enabled applications because of their decentralised data storage characteristics, anonymity, data protection and access control of their EMR and EHR data [52] [80]. In addition, the use of blockchain can expand the current Personal Health Record (PHR) data management

system to combine event-driven smart contracts to support transactional services such as repeat prescriptions, booking appointments, and requests for referrals [80]. Others have deployed a blockchain-enabled decentralised app (DApp) and platform to tackle the interoperability issues in health care facilities, allowing patients to use the DApp to exchange their clinical details as the basis for remote support decision-making [43] [163]. Nevertheless, most of the clinical environments lack real-world use cases.

However, these schemes can provide secure storage and efficient access control but fail to prevent insider attacks and cloud server crashes. Most of the existing approaches do not guarantee all the essential requirements for EHRs, such as data privacy, security, secure storage, effective access control, scalability and interoperability. Our research work addresses most of the current e-health challenges by using a permissioned blockchain platform through the use of Practical Byzantine Fault Tolerance (PBFT) as a consensus to allow data sharing in a decentralised fashion through IPFS by maintaining effective patient privacy, confidentiality and health record integrity. From the comparative study in Table 5.1, it is clear that the proposed framework resolves most of the issues with the existing techniques and offers a foolproof solution to e-health data implementations.

5.3 Proposed Methodology

The proposed architectural overview is portrayed in Fig. 5.3. This framework comprises stakeholders or participants, Angular 4 application, Fabric SDK, Hyperledger Composer, Hyperledger Fabric, Chaincode, CouchDB and IPFS. Angular 4 is the Front end of the DApp framework that connects with the Composer Rest server which exposes and visualizes the state database, couchDB. This application consists of four types of users namely doctors, patients, chemists and receptionists with n participants for each user. The Fabric-CA provides key public certificates for all n applicants, including patients, doctors, receptionists and pharmacists. The Membership Service Provider abstracts all the cryptographic mechanisms such as identity validation, signature

generation and verification, certificate issuance and validation protocols and healthchain user authentication. Users can interact with the main application via the Angular 4 user interface. User can send and invoke queries via Fabric SDK. SDK will verify the global state of the blockchain and submit a query to the blockchain via the Composer restful service-based API. Healthchain will also send the request to other peers for consensus. After the successful consensus, the transaction will be submitted to the blockchain and the subsequent key-value pair will be created or modified according to the request. The

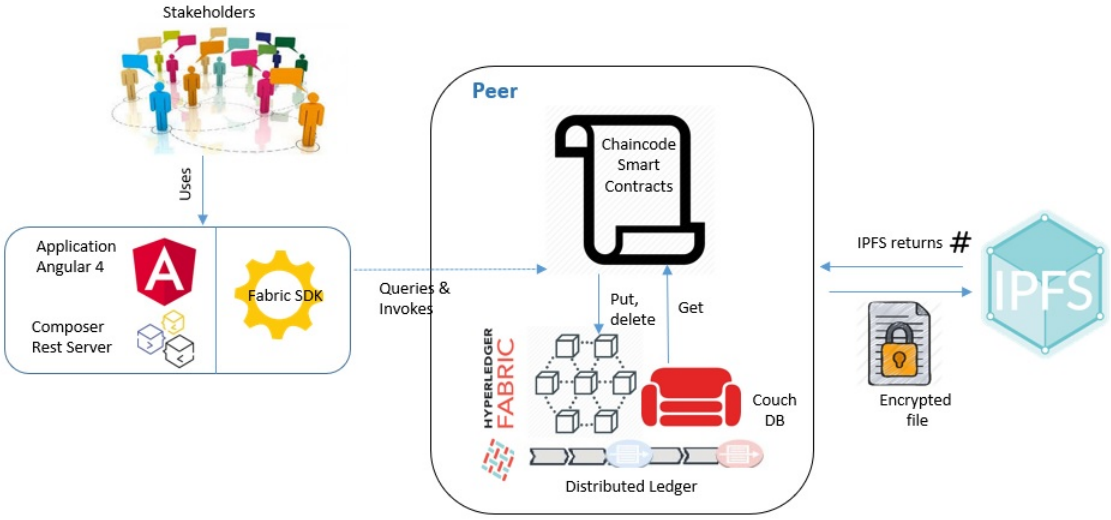


Fig. 5.3 Overview of Workflow in the Healthchain network.

REST API is used to get the actual state of the couchDB chain database in which the angular frame retrieves data through GET calls to the REST API of the composer. Hyperledger Fabric is the underlying permissioned blockchain technology for distributed ledger solutions that support the building of chaincodes known as smart contracts written in Go, Node.js to validate medical data entries and transactions in the health data network. Whenever a user logs into the application, the credentials are verified from the back end REST API. Now with every query, the application passes the user credentials on the backend REST API verifies the identity of the person. This is the first layer of security. When actual data is being pulled from the blockchain, the blockchain verifies the user identity in the state database via REST API when the user was created on the blockchain which forms the second layer of security. Any communications with medical records are recorded as network transactions and only the parties participating in the transaction will see the behaviour of the transaction.

The working prototype implemented a private blockchain on Hyperledger Fabric called Healthchain by using Hyperledger Composer to create decentralised web applications for a single organisation by incorporating a single node. This organization has one peer node (validating node) and an ordering node with a single public channel for registering the network participants. The system contains a single peer node configured to use CouchDB as the world state database and IPFS as the distributed database, a solo ordering node, a Certificate Authority, Membership Service Provider (MSP) and smart contracts for connecting to the blockchain. This research uses Hyperledger Fabric as the authorised blockchain framework consisting mainly of pre-specified parties for the reliable and secure sharing of health information without any central authority [8]. The greatest advantage of this research is that it uses Byzantine fault tolerant consensus protocol that does not involve mining or an associated currency to achieve consensus. One primary property and fundamental layer of blockchain is the consensus mechanism for transactions that depend on the layer of smart contracts to validate and update transactions in the ledger according to the order in which they occur. The Consensus Protocol establishes the order and rejects bad transactions in the ledger. Practical Byzantine Fault Tolerance (PBFT) is the consensus hard-coded in this system, which uses either crash tolerant or Byzantine fault tolerant and does not require mining to achieve consensus [8]. CouchDB is the on-chain database used in this research that can also improve compliance, security and data protection in the Healthchain which can be validated by querying the Composer REST server. Composer in this research creates a business network definition comprised of model file(.cto), script file(.js), ACL(.acl) and Query(.qry) files and it packages the business network definition to a business network archive (.bna) file for deployment in the Healthchain business network to a distributed ledger [45]. This research work uses smart contracts that encompass the application logic of the system for EHR transactions particularly for eReferrals between clinicians, data transmission, access management, request handling such as update medical records, update ownerships etc. Smart contracts will be executed during user interaction to identify request, validate request, secure clinician interaction, for granting access permissions and update permissions for medical records. IPFS is used in this research as an off-chain database for the storage

and encoding of infinite healthcare records using a public key encryption before storage and hash of the records will be stored in couch database that models the database of our framework [17]. A snippet of the scripting component code used in eReferral is shown in Appendix B.3.

Table 5.2 Explanation of Notations.

Notations	**Definition**
IPFS	InterPlanetary File System
P_{Cv}	Composite data view
P_L	Patient Ledger
C_L	Clinician Ledger
N_{Adm}	Network Admin
H_N	Healthchain Network
C_{ID}	Clinician ID
S_k	Session Key
C_{Pk}	Public Key of Clinician
C_{Pr}	Private Key of Clinician
P_{EHR}	Patients' Health record
P_{Pk}	Patients' public key
P_{Pr}	Patients' private key
P_i	Patient
C_i	Clinician
R_i	Receptionist
Ph_i	Pharmacist
UP_{Cv}	Updated Composite view
UP_{EHR}	Updated Health Record

5.3.1 Cryptographical Process in eReferral

Fig. 5.4 demonstrates the cryptographical process of providing access permission and key generation for the referred clinician. This research employs public key encryption for securing data in the off chain database IPFS and the comprehensive approach is outlined in Fig. 5.4. The authorized clinician checks whether a referral is required and sends the encrypted referral report to the specialist. The referred clinician (doctor) requests patient approval to access the patient's additional health record stored in the IPFS. The patient approves the request of the permissioned users on the basis of the

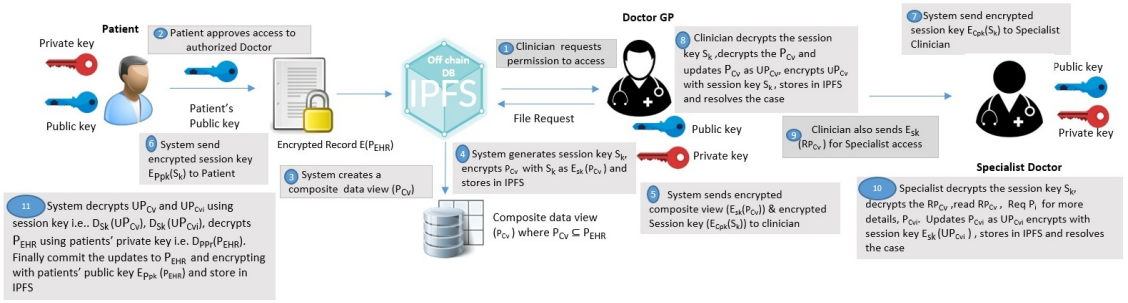


Fig. 5.4 Cryptographical Mechanism in HealthChain.

access control permission rules. The system in this framework refers to the client-side application. The system further generates a composite view of the health record upon request without sharing the whole patient data. Composite view P_{Cv_i} is the attribute set of the stored medical record P_{EHR_i} that the system creates on permissioned user request without sharing the complete patient record. The composite view of a specific health record restricts access to the original data in such a way that a user can only see and modify the selected data they need and no more. The system generates a session key S_k to access records for a definite session and encrypts the composite view with the session key and then stores in IPFS. The system will also send the encrypted session key and encrypted composite view to the referred clinician. Furthermore, the system also shares the encrypted session key with the patient. The referred clinician decrypts the session key, decrypts the composite view and updates the composite view as an updated record. Further, the referred clinician resolves the instance after encrypting the updated record with the session key and uploads it to the IPFS. The system decrypts the updated composite view using the session key, decrypts the encrypted medical record with patient's private key from the IPFS. Finally, the patient commits the updates to the original record, encrypts the original record with the public key of the patient and uploads it to the IPFS. The session key and the composite view for each session expires on session completion. The detailed explanation with notations is illustrated in the proposed algorithms.

5.3.2 Transaction Workflow in eReferral

Fig 5.5 illustrates the work flow of eReferral in the Healthchain framework and Fig 5.8 describes a snippet of the smart contract employed for eReferral in the Healthchain framework. Assuming the stakeholders in Fig 5.5 are registered participants, initially the doctor(clinician) and the patient log in with their credentials to the permissioned

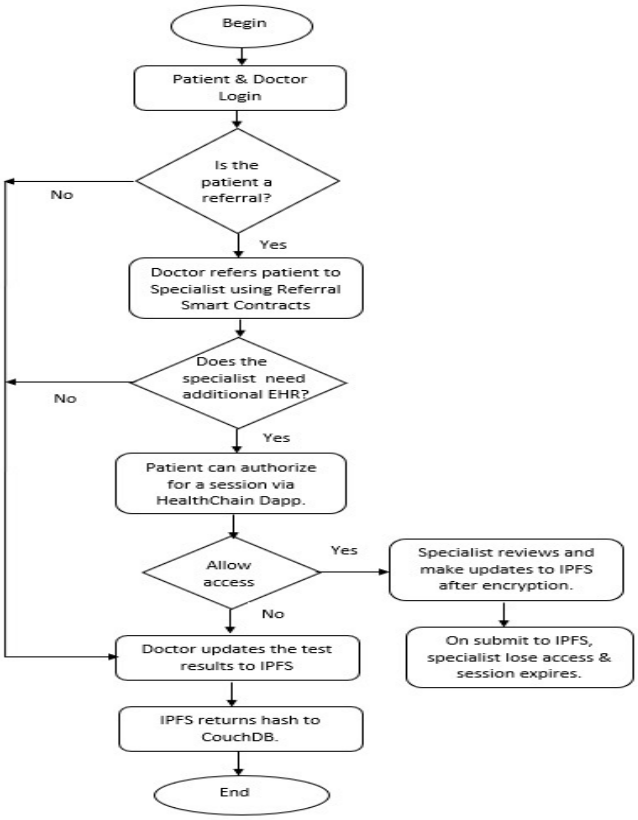


Fig. 5.5 Overview of Workflow in the Healthchain Network for eReferral. EHR:Electronic Health Record; Dapp: Decentralised Application; IPFS:InterPlanetary File System; CouchDB: Couch Database.

blockchain. The authorized doctor (clinician or general practitioner) checks whether a referral is needed and refers the patient to specialist practitioner with the required patient details. The referral is performed via employing the eReferral smart contract functionality. The specialist clinician can request for additional health information from the patient via Healthchain Dapp. The referred doctor can access the EHR for a particular session once the patient approves access to his/her patient details. The specialist reviews and makes updates if required and uploads the record to IPFS after

encrypting the record with the associated session key. The specialist loses access to the patient's details and the session expires on task completion. However, if the patient is not a referral case, the doctor performs a normal patient assessment, forms a diagnosis, administers patient care and uploads the test results to IPFS. The IPFS returns a hash for each transaction and stores the value in couchDB.

Fig. 5.6 shows the step-by-step details of adding medical records to the Healthchain by the referred clinician. This approach begins with assuming that the patient and the clinician have established an authorized relationship for updating health records. The

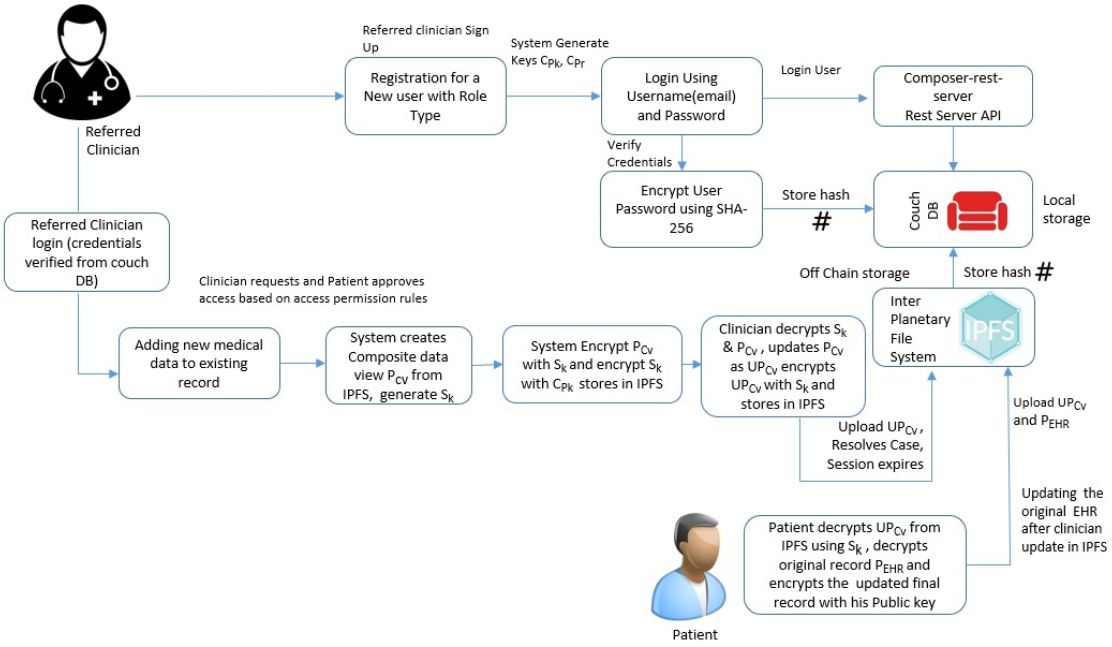


Fig. 5.6 Illustration of Adding Records to Healthchain.

process of adding medical records by the referred clinician to the database is employed via the internal encryption mechanism. The referred clinician will be added to the healthchain using their credentials such as username and password with each user having public private key pairs Pk_i, Pr_i. The user password is encrypted using the SHA-256 hashing algorithm for improved security. A new patient record will be inserted or updated by the referring clinician following receipt of the referral documentation from the GP and after review. If the specialist requires additional information, the system creates a composite view, P_{Cv_i} of the data that is accessible by the clinician SC_i alternately sharing the whole data. The system further generates a

```
<?xml version="1.0"?>
<Access-control-rules>
    <role_name> Clinician Ci </role_name>
    <permissions desc="Permissioned clinician
     authorized by the Patient">
     <resouce desc="EHRi"> Electronic Health Record
</resouce>
     <Object>
         <object.id=Ci.id>
     </Object>
     <action type="write"> ALLOW </action>
     <access mode>
        <access.mode="normal">
     </access mode>
     </permissions>
     .......
     <role_name> Specialist Clinician SCi </role_name>
     <permissions desc="Permissioned referred
   clinician authorized by the Patient">
     <resouce desc="RPi"> Referral Report </resouce>
     <Object>
         <object.id= SCi.id>
     </Object>
     <action type="read"> ALLOW </action>
     <access mode>
        <access.mode="normal">
     </access mode>
     </permissions>
     <role_name> Specialist Clinician SCi </role_name>
     <permissions desc="Permissioned clinician
    authorized by the Patient">
     <resouce desc="EHRi"> Electronic Health Record
</resouce>
     <Object>
         <object.id= SCi.id>
     </Object>
     <action type="write"> ALLOW </action>
     <access mode>
        <access.mode="normal">
     </access mode>
     </permissions>
     .......
     </Access-control-rules>
```

Fig. 5.7 Access Control Rules for eReferral.

session key S_k shared by the patient and the clinicians for a distinct session. The system then sends the encrypted session key S_k to the patient as $E_{P_{pk_i}}(S_k)$ and specialist clinician as $E_{C_{pk_i}}(S_k)$ by encryption using the respective public keys of the patient P_{Pk_i} and clinician SC_{Pk_i} for a distinct session. The Composite view P_{Cv_i} will also be encrypted with session key S_k as $E_{S_k}(P_{Cv_i})$ and stores it in IPFS. In addition, the system sends an encrypted composite view i.e. $E_{S_k}(P_{Cv_i})$ to the clinician. Now, the clinician decrypts the session key with his private key and decrypts the composite view with the session key. If there are any updates, the clinician updates P_{Cv_i} as UP_{Cv_i}, resolves the case, encrypts it with the session key and uploads UP_{Cv_i} to IPFS as $E_{S_k}(UP_{Cv_i})$. On the clinicians' record update, the system decrypts the encrypted record i.e. $E_{P_{Pk_i}}(P_{EHR_i})$ using the patients' private key and also decrypts the encrypted updated composite view

from the IPFS i.e. $E_{S_k}(\mathrm{U}P_{Cv_i})$ using the session key. The patient uses a pass code to encrypt the private key P_{Pr_i} and stores it on the client side. Each time, the patient can supply this passcode to decrypt the private key rather than exchange or upload the private key, and this private key can be used by the end-user application to decrypt the medical record. Finally, the patient commits the updates to the original record and encrypts the original record P_{EHR_i} as $E_{P_{Pk_i}}(P_{EHR_i})$ before uploading it to IPFS. The session key S_k for each session expires and the composite view P_{Cv_i} will be deleted after the session is completed. The transactions eventuated on the clinician's access and record updates will be hashed by employing smart contracts and added to the healthchain. This procedure is summarized in Algorithms 3 and 4 and Algorithm 5 for access management as shown in Fig. 5.7 .

5.3.3 Proposed Algorithms

This approach starts with the assumption that the patient and the clinician have formed an authorised relationship to update health records. This framework has four stakeholders, such as doctor (clinician), patient, specialist clinician and the receptionist with n users for each participant. There are 3 algorithms in which Algorithm 1 illustrates the patient working in the network, Algorithm 2 illustrates the clinician working and Algorithm 3 illustrates the specialist clinician working in the Healthchain network . Table 5.2 explains the notations used in the algorithm. The patient has read access to their own health records and can provide read, write, revoke and deny access permissions to the authenticated stakeholders in the network. The process of adding medical records to the database by the clinician is undertaken via an internal encryption mechanism as shown in Fig.5.4. The referred clinician will be able to view the referred details by the general practitioner. If P_{EHR_i} is not in the network, then the patient provides the clinician with access to create P_{EHR_i}. For an existing record upon the clinician's request, the system creates a composite view P_{Cv_i} of the patient record P_{EHR_i}, alternately sharing the whole medical record of the patient as shown in step 15 of Algorithm 3. Composite view P_{Cv_i} is the attribute set of the stored medical record P_{EHR_i} that the system creates on the permissioned user request without sharing the

Algorithm 3 :Algorithm on Patient working

Input: P_{ID} and P_{Pk}

Output: Get Access to Patient ledger transactions $P_L \in H_N$

Initialisation : P_L should be a valid node and can Read, Revoke, Grant or Deny EHR records

procedure Patient (P_{ID})
while (True) do
if ($P_{ID} \in H_N$) **then**
  if ($P_{EHR_i} \notin H_N$) **then**
    create_records(P_{ID}, P_{EHR_i}, H_N)
  else
    read_records(P_{ID}, P_{EHR_i}, C_{ID}, H_N)
  end if
else
  P_{ID} is invalid
end if
if visit (P_{ID} ,C_{ID}, H_N) **then**
  P_{EHR_i} = Medical_record (P_{ID})
  if ($P_{EHR_i} \in P_L(H_N)$) **then**
    $P_{CV_i} \leftarrow \int_{i=1}^{n} (D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i})))$
    Grant_records(P_{CV_i}, C_{ID}, S_k, H_N) where $P_{Cv_i} \subseteq P_{EHR_i}$
    $C_i \leftarrow E_{C_{Pk_i}}(S_k)$
    $C_i \leftarrow E_{S_k}(P_{Cv_i})$
    Algorithm 4 ()
  else
    (C_{ID}) $\leftarrow$ NOTIFY ("Medical records does not exist")
  end if
  if (UP_{Cv_i}) **then**
    $P_{EHR_i} \leftarrow [(D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i}))) + E_{P_{Pk_i}}(UP_{Cv_i})]$
  end if
  if ($P_{EHR_i} \in C_{ID}$, treatment completed (P_{ID})) **then**
    end session (S_k, P_{EHR_i}, C_{ID})
  else
    (C_{ID}) $\leftarrow$ NOTIFY ("voluntary revoke P_{EHR_i}")
    Revoke_records(P_{EHR_i}, C_{ID}, H_N)
  end if
else
  Not visit
end if
end while
end procedure

Algorithm 4 :Algorithm on Clinician working

Input: C_{ID} and C_{Pk}

Output: Get Access to Clinician ledger transactions $C_L \in H_N$

Initialisation : C_L should be a valid node and can Read or Write EHR records permissioned by the patient

procedure Clinician (C_{ID})
while (True) do
if ($C_{ID} \in H_N$) **then**
  if (Granted P_{EHR_i}) **then**
    Read_records(C_{ID}, P_{EHR_i}, H_N)
    Update_records (C_{ID}, UP_{Cv_i}, H_N)
    $C_i \leftarrow D_{C_{Pr_i}}(S_k)$
    $C_i \leftarrow D_{S_k}(P_{Cv_i})$
    $RC_i \leftarrow E_{S_k}(RP_i)$
    $P_{Cv_i} \rightarrow (UP_{Cv_i})$
    IPFS $\leftarrow E_{S_k}$ (UP_{Cv_i})
  end if
  Specialist $\leftarrow RP_i$
  Algorithm 5 ()
else
  C_{ID} is invalid
end if
end while
end procedure

Algorithm 5 :Algorithm on Specialist Clinician working

Input: SC_{ID} and SC_{Pk}

Output: Update Records to Patient Ledger transactions $P_L \in H_N$

Initialisation : SC_{ID} should be valid and can Read and Write Medical records permissioned by the Patient

procedure Specialist Clinician (SC_{ID})
while (True) do
if ($SC_{ID} \in H_N$) **then**
  if (Granted RP_i) **then**
    Read_records(SC_{ID}, RP_i, H_N)
    Specialist Clinician $\rightarrow$ (RP_{Cv_i})
    IPFS $\leftarrow E_{S_k}$ (RP_{Cv_i})
  end if
  Algorithm 3 ()
else
  SC_{ID} is invalid
end if
end while
end procedure

Algorithm 1: Smart Contract for Patient records

```
Assign Roles to stakeholders
function Define Roles (New role, New Account)
    Add new role and new account in Healthchain
    access based on access control permission rules
end function
function create medical record (contains asset variables to create record)
    If (msg.sender_id == GPdoctor_id) then
        Create medical record to patient's record
        IPFS ←— E_Sk (UP_Cvi) /*Encrypts updated composite view with Clinician's session key
        return hash #          /* Return hash value to CouchDB, blockchain
    else Abort session
    end if
end function
function create patient referral record (contains asset variables for referral)
    If (GPdoctor_id == doctor_id && referdoctor_id== doctor_id) then
    if (patient_id ==true && record_id==true)
            return record from specific patient_id
            create patient referral record
            update data to particular patient's record
            return hash #
     else Abort session
     end if
     end if
end function
function view patient record (patient id)
     if (msg.sender_id == doctor || patient) then
     if (doctor_id == true && patient_id==true) then
            return patient record
     else Abort session
     end if
     end if
     end function

function update patient record (contains asset variables to update patient record)
     if (msg.sender_id == doctor) then
     if (doctor_id == true && patient_id==true) then
            P_EHRi ←— [( D_PPri (E_PPki (P_EHRi ))) + E_PPki (UP_Cvi)]   /* Store updated medical record to IPFS
            IPFS ←— UP_EHRi update patient record
            return hash #
     else return fail
     end if
     else Abort session
     end if
 end function
```

Fig. 5.8 Snippet of Smart Contract for eReferral.

complete patient record. In other words P_{Cv_i} is a subset of P_{EHR_i} as shown in equation 5.1.

The system further generates a session key S_k shared by the patient and the referred clinician for a distinct session. The system then sends the encrypted session key S_k to the

patient as $E_{P_{pk_i}}(S_k)$ and clinician as $E_{C_{pk_i}}(S_k)$ by encryption using the respective public keys of the patient P_{Pk_i} and clinician C_{Pk_i} for a distinct session as shown in step 17 and 18 of Algorithm 3. The composite view P_{Cv_i} will also be encrypted with session key S_k as $E_{S_k}(P_{Cv_i})$ and stored in IPFS. In addition, the system sends an encrypted composite view i.e. $E_{S_k}(P_{Cv_i})$ to the clinician. The clinician decrypts the session key with his private key and decrypts the composite view with the session key as shown in steps 7 and 8 of Algorithm 4. If there are any updates, the clinician updates P_{Cv_i} as UP_{Cv_i}, resolves the case, encrypts it with the session key and uploads UP_{Cv_i} to IPFS as $E_{S_k}(UP_{Cv_i})$. The clinician also sends patient referral $E_{S_k}(RP_i)$ if required to the specialist clinician. The specialist decrypts the associated session key S_k, decrypts and reads the (RP_i), and requests more details from patient P_i. The process of generating a composite view repeats and the specialist updates P_{Cv_i} as UP_{Cv_i} encrypts it with session key $E_{S_k}(UP_{Cv_i})$, stores it in IPFS and resolves the case. On the clinicians' record update, the system decrypts the encrypted record ie. $E_{P_{Pk_i}}(P_{EHR_i})$ using the patients' private key and also decrypts the encrypted updated composite view from the IPFS i.e. $E_{S_k}(UP_{Cv_i})$ using the session key as shown in eqn(5.2). Finally, the system commits the updates to the original record and encrypts the original record P_{EHR_i} as $E_{P_{Pk_i}}(P_{EHR_i})$ before uploading it to IPFS as shown in equation (5.3). The session key S_k for each session expires and the composite view P_{Cv_i} will be deleted upon session completion. The transactions eventuated on the clinician's access and record updates will be hashed by employing smart contracts and added to the Healthchain ledger.

$$P_{Cv_i} \subseteq P_{EHR_i} \tag{5.1}$$

$$P_{Cv_i} = (D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i}))) \tag{5.2}$$

$$P_{EHR_i} = [(D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i}))) + (E_{P_{Pk_i}}(U_{P_{Cv_i}}))] \tag{5.3}$$

5.4 Implementation Results

Fig. 5.9 and Fig. 5.10 show the process of referring health records for the doctor's referral by employing unique attributes in the Healthchain. Fig. 5.9 illustrates the medical record creation, uploading of the health record in to IPFS, the hash generation in the doctor's profile and also shows the querying and retrieval of the record details through Composer. This shows how a new medical record is created in the doctor's profile by employing unique recordId, patientId, doctorId, file description, encounter time and location in the healthchain network. It can also be seen that the system generates a unique hash for the uploaded file. The details of the created record can be retrieved by querying the Composer as shown in the figure. A snippet of the medical record creation scripting is shown in Appendix. B.2. Fig. 5.10 shows how the referral records are created in the

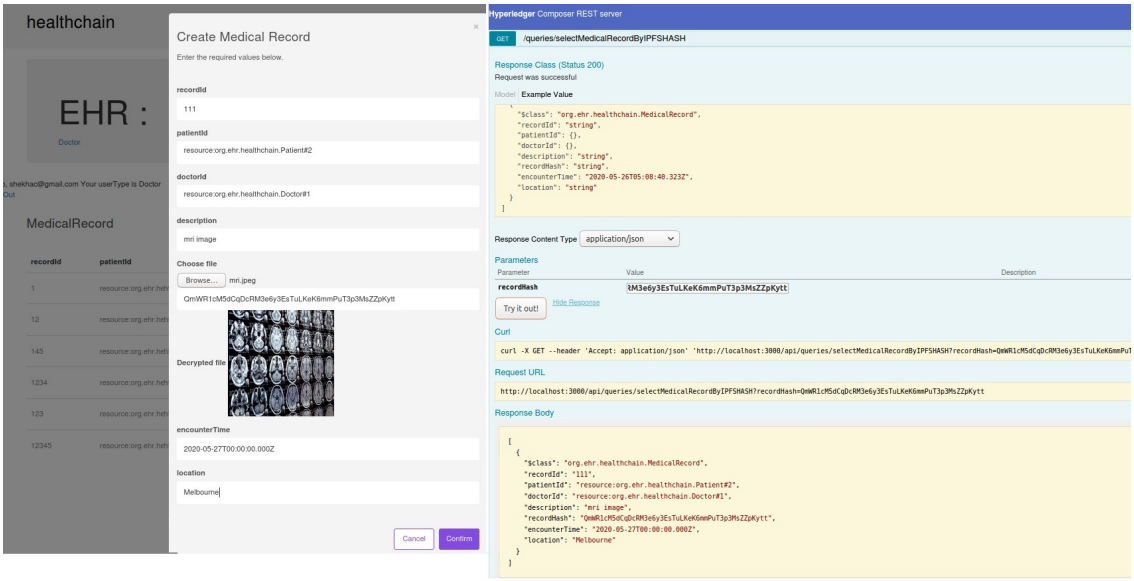


Fig. 5.9 Illustration of EHR Record Creation and Hash Generation in the Referred Doctor's Profile and Retrieving the Record Details through Composer.

doctor's profile. An asset patient referral record will be created in the GP doctor's profile employing unique attributes such as recordId, patientId, GPdoctorId, referred doctorId and referral description. The referral document can be then accessed from the referred doctor's profile.

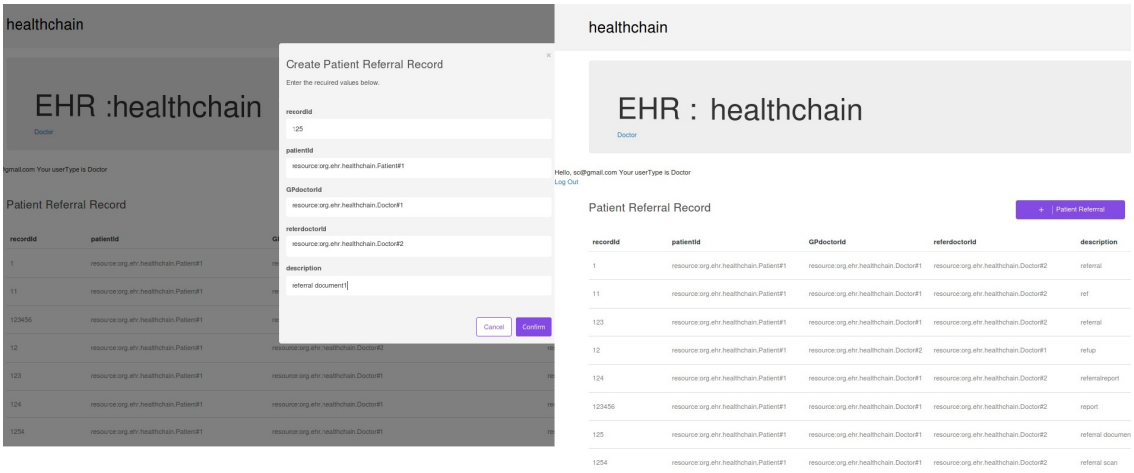


Fig. 5.10 Illustration of Referral Records in Healthchain.

5.4.1 Case Study and Analysis

A few test cases have been conducted to evaluate the framework's feasibility and system performance.

- Case I : Efficient Creation of Health Records
- Case II : Efficient Creation of Referral Records
- Case II : Effective Security and Access Permissions

Efficient Creation of Health Records- Case I : The efficient creation of health records in the Interplanetary file system and the referred clinician's profile is tested against a few cases listed in Fig 5.11. The first test case verifies if a referred doctor can upload medical records or update test results on IPFS. The implementation results shown in Fig. 5.9 shows that the specialist doctor authenticated by the patient can have write access to the medical records and upload encrypted records into IPFS. The second case is tested if the referred doctor has read access permission and can view the referral records and is successfully verified as the doctor who has been authenticated by the patient. Furthermore, it tests that a patient can view the referral records created by the referred clinician and Fig. 5.10 portrays the provenance history of the medical records. Moreover, the system is tested against whether a referral record can be uniquely identified or not and has been successful as every medical record is uniquely related with a record id, doctor id and patient id. Additionally, the system has been checked to

S.No	Test case	Description	Outcome
1	Verify if a referred doctor can upload medical records on IPFS	The specialist doctor authenticated by the patient can have write access for the record and upload encrypted records into IPFS	Passed
2	Verify if a doctor can view the referral records on permission	The specialist doctor authenticated by the patient can have read access for the referral records	Passed
3	Verify if a patient can view the referral records	The patient can see the provenance history of the records	Passed
4	Verify if a referral record is uniquely identified	Each referral record is uniquely associated with a recordId, patient Id and doctor id	Passed
5	Verify if an encrypted record can be effectively retrieved	The updated medical record can be encrypted with specialist doctor's session key for storing in IPFS and updated record can be decrypted by using patients' session key at the patient side	Passed

Fig. 5.11 Efficient Medical Record Creation in Healthchain.

see whether an encrypted record can be effectively retrieved after decryption and has been successful as shown in Fig.5.9 in the decrypted file component. The outcome is successful as the updated record can be encrypted with specialist doctor's session key for storing in IPFS and the updated record can be decrypted using the patients' session key at the patient side.

Efficient Creation of Referrals- Case II : Referral report creation in the Healthchain has been verified against a few test cases, as shown in Fig. 5.12. The first case is tested and successful as the specialist doctor authenticated by the patient can create referrals. The second test case verifies that the referred doctor can view the referrals created by any doctor. The outcome is favorable as the specialist doctor authenticated by the patient can view and read the referrals. Furthermore, the prototype has also been verified as to whether the patient can view all the referral records and is found successful. The prototype has also been tested to check whether record details can be retrieved and has been successful as shown in Fig. 5.9. The prototype has also been tested as to whether the referred clinician can update the medical records to IPFS and has been found successful for a particular session.

Effective Security and Access Permissions- Case III : Data privacy and security in eReferral mechanism have been checked against a few use cases as shown in Fig. 5.13. The initial case is verified and successful as the general practitioner can provide referral

S.No	Test case	Description	Outcome
1	Verify if a doctor can create patient referral report	The doctor authenticated by the patient can create referral record employing unique recordId	Passed
2	Verify if the referred doctor can view the referral records created by any doctor	The specialist doctor authenticated by the patient can have read access for the referral records created by any doctors	Passed
3	Verify if the patient can view the referral record details	The patient can see the provenance history of the referral records	Passed
4	Verify if a referral record details can be retrieved	Each referral record is uniquely associated with a recordId and can be retrieved	Passed
5	Verify if a referral record can be effectively retrieved from the referred doctor profile and referred doctor can update medical records in IPFS	Referral record can be retrieved from referred doctor profile and the updated medical record can be encrypted with specialist doctor's session key for storing in IPFS and updated record can be decrypted by using patients' session key at the patient side	Passed

Fig. 5.12 Efficient Creation of Referral Records.

reports for a particular session after encrypting it with the session key. Additionally, the system has been tested to check whether a patient can provide grant access, revoke access and permit access permissions of the health records to the stakeholders and has been successful in preserving data privacy. Furthermore, the system is also tested to see whether the specialist clinician can upload updated records to IPFS and the result is successful as the referred clinician utilizes session key encryption before the session expires. Finally, the prototype is checked to see whether the specialist doctor can view the record using the security token and the outcome is successful as some session tokens has been added to the framework and they expires on session completion.

S.No	Test case	Description	Outcome
1	Verify if a clinician can provide secure referrals to the specialist clinician in healthchain	Clinician sends encrypted referrals after session key encryption for specialist access for a definite session	Passed
2	Verify if a patient can grant access ,revoke access, and provide access permission of medical record with the specialist clinician	The patient has all the permissions to grant or revoke access from other user types to maintain privacy and patient has the permission to provide read, write, and deny access permission according to the role type	Passed
3	Verify if the specialist clinician can upload additional details of records in IPFS	The referred clinician encrypts the record with session key and upload to IPFS before the session expires	Passed
4	Verify if the specialist doctor can view the medical record using security token	Since the patient can revoke access of their medical record from the doctor, providing access again is troublesome and hence some session token has been added into the framework that expires after the session	Passed

Fig. 5.13 Effective Security and Access Permissions.

5.4.2 Empirical Analysis

Several experiments have been carried out to analyse and evaluate the performance of the proposed healthchain network. The assets defined here are: (a) Medical Record (b) Referrals (c) Prescription (d) Add Ownership. The transactions are: (a) Create Medical Records (b) Update Medical Records (c) Allow Doctors Write (d) Update Ownership (e) eReferrals to other Doctor (f) ePrescription to Pharmacist. This research evaluated the transaction latency, transaction throughput and time latency for asset creation while sharing the eReferral between stakeholders in the Healthchain network.

Transaction latency is the amount of time taken for the transaction to commit and become available across the peer nodes in the network. If there are n number of nodes in the Healthchain network, T_{L_n} is the transaction latency, T_{C_n} is the confirmation time in the network nodes and T_{S_n} is the transaction submit time in seconds then;

$$T_{L_n} = T_{C_n} - T_{S_n} \tag{5.4}$$

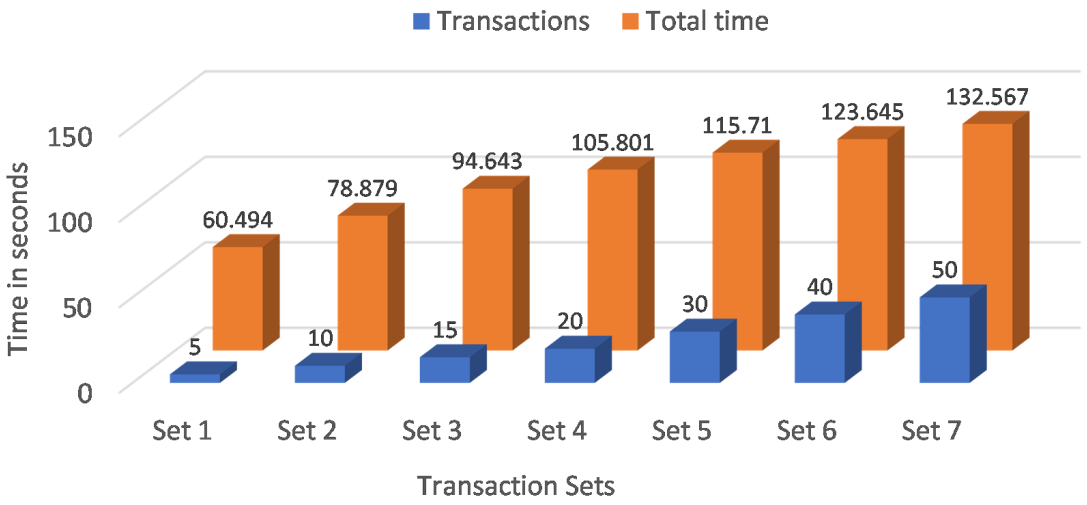


Fig. 5.14 Transaction Latency.

In this chapter, transaction refers to sharing the eReferral securely to authenticated clinicians. Here, we have experimented on seven sets of transactions within a range of 5, 10, 15, 20, 30, 40 and 50 for updating to the network ledger as shown in Fig.5.14.

Considering the machine configuration in Table 4.3, it is evident that the initial set of 5 transactions took an average of 60 seconds to commit across the network and the final set of 50 transactions took an average of 130 seconds.

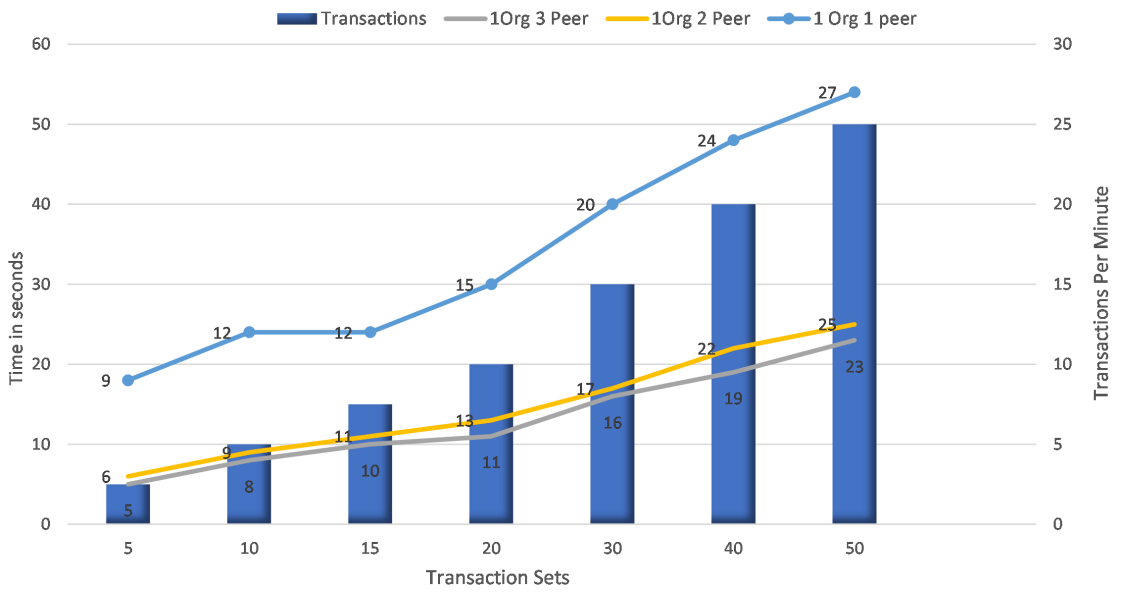


Fig. 5.15 Transaction Throughput: Comparative Analysis.

The result obtained is then analysed for a comparative study of transaction latency in 1 Org 1Peer, 1 Org 2Peer and 1 Org 3Peer as shown in the Fig. 5.15. The second experiment calculated the transaction throughput or transactions per minute (TPM) for various sets of transactions of the proposed framework. The transaction throughput is the rate at which the blockchain system under test (SUT) commits valid transactions in a defined time period at all network nodes. If there are n number of nodes in the blockchain network, T_{T_n} is the transaction throughput, T_{Ct_n} is the total number of committed valid transactions in the network nodes and T_{Tot} is the total time in seconds then:

$$T_{T_n} = T_{Ct_n}/T_{Tot} \tag{5.5}$$

For seven sets of transactions ranging from 5, 10, 15, 20, 30, 40 and 50, it is clear that TPM for 1 Org 3Peer is lower than TPM for 1 Org 1Peer as shown in the Fig. 5.15. Therefore, this shows that the higher the number of peers, the lower the number of valid transactions across the network. The asset latency is the time taken by the SUT to

successfully load and write the assets to the couchDB. Here, creating a patient referral record can be considered as an asset. If there are n number of nodes in the blockchain network, A_{L_n} is the asset latency, T_{Res_n} is the response time and T_{Sub_n} is the asset submit time in milliseconds then:

$$A_{L_n} = T_{Res_n} - T_{Sub_n} \tag{5.6}$$

Fig. 5.16 shows varying asset sizes in bytes of 5 concurrent users in three nodes in the

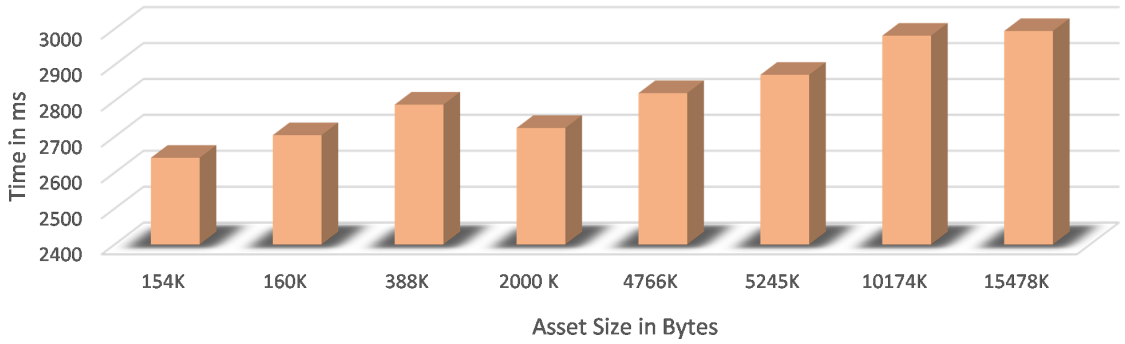


Fig. 5.16 Asset Latency.

proposed system and it can be seen that it takes an average latency of 3.0 seconds to commit asset write updates in the couchDB across the network. It is observed that an asset size of 154K bytes takes an average of 2.6 seconds and 15478K byte size takes an average of 3.0 seconds to commit write updates in the CouchDB. This work can be extended to n number of nodes and a different number of organisations to test system feasibility can be considered as future work.

5.5 Summary

In this research work, a permissioned blockchain framework is implemented for secure data storage and access to electronic health records utilizing Hyperledger Fabric and

Hyperledger Composer. Since the blockchain is tamper-resistant, the system is tamper-proof and can handle healthcare records while preserving data privacy, security and integrity. Moreover, no incentive mechanisms for blockchain mining are included that demonstrates the patients' ownership of their healthcare data. The presented framework and the results of the prototype based on the test cases can be summarized as follows:

• This research provides a Distributed Ledger Technology Smart contract system for efficient eReferral between multiple clinicians in the healthdata network in the medical industry. This work creates smart contracts for various medical workflows, and then data access permissions are managed by the patient in the healthcare ecosystem.

• This research proposes an architecture for securing data storage and providing efficient access control between stakeholders viz patients, doctors, pharmacists and other participants via encryption techniques and access control mechanisms.

• A working prototype based on Hyperledger Fabric and the Interplanetary File System is made to illustrate the system's viability. The proposed methodology is implemented and evaluated with some use cases for EHRs. Consequently, the framework is successful as a reliable health data network.

• The results of prototype implementation and analysis prove that the approach is a tamper-resistant mechanism as information will be stored as hash values for every healthcare transaction in the blockchain. Moreover, it has enormous potential to ensure the privacy, security, integrity, confidentiality and scalability of e-health information.

• This research also explores the technology framework and business processes for blockchain applications.

Chapter 6

Pharmaceutical Supply Chain Integrity Management and Provenance using the Healthchain Framework

The healthcare field is facing a major problem of prescription drug abuse or doctor shopping that entails drug misuse, leading to the fatality of a large number of people worldwide. Painkillers like oxycodone and vicodin which are over-prescribed by doctors are among the most abused legal drugs alongside sleeping pills and anxiety medication. For this reason, there is an imminent need to devise a system that identifies and monitors prescription abuse. Blockchain technology's decentralisation and auditability offers a promising solution to drug tracking that not only makes prescriptions safer but also guarantees a reliable transaction history of medical records. Blockchain is one of the best ways to ensure the transparency, integrity and authenticity of the pharmacist's or doctor's distribution of drugs. This chapter *(i) proposes a novel drug supply chain integrity management system using blockchain technology by employing Hyperledger Fabric as the underlying blockchain platform and the InterPlanetary File System (IPFS) as the decentralised file system to prevent prescription abuse as it guarantees precision with its secure cryptology framework and safeguards against fraud and forgery; (ii) solves this problem by performing drug*

tracking transactions by employing smart contract functionality on a blockchain to create a smart health care ecosystem. Also, through this approach, it is possible to recognize and track doctor shopping and pharmacy hopping patients who may be attempting to misuse drugs. Healthchain seeks to improve the way opioid and prescriptions are administered and distributed by creating a cryptographically secure and reliable framework for physicians, pharmacists and patients.

6.1 Introduction

Doctor shopping is the process of visiting many physicians without a professional referral to receive several prescriptions for drugs, or the medical opinion one needs to hear [22] [38]. It has serious consequences for patients, as numerous consultations and overlapping prescriptions are related to drug abuse, polypharmacy, rising medical expenses and increased mortality rates. There are several explanations as to why patients engage in doctor shopping. Patients see a number of doctors when they have a chronic disease or they are engaged in drug abuse or after seeking medication, their health condition remains unresolved. This is a common practice for drug addicts, drug addiction suppliers, hypochondriacs or factitious disorder patients. These medications assist the patient to obtain immediate pain relief, but they have also disadvantages, despite the advantages. The current prescription opioid marketplace is riddled with data hoarding, doctor shopping, provider ignorance, vulnerabilities, centralized data, and over-prescription. According to the statistics, an estimated 237m drug mistakes occur annually and the expense of avoidable adverse effects is calculated at £98.5 million a year, taking 181,626 bedding days, resulting in 712 deaths and 1,708 mortally afflicted in NHS England [47]. Moreover, according to a study in the CURES (California's de-identified Controlled Substance Utilization Review and Evaluation System) dataset, 10% of random samples included 17,954,968 opioid prescriptions written by 185,424 prescribers to 3,044,579 patients with some predominated opioid [126]. Fig.6.1 demonstrates how opioid overdoses have increased over time since the beginning of the opioid epidemic. Since opioid misuse is a challenging problem that requires a

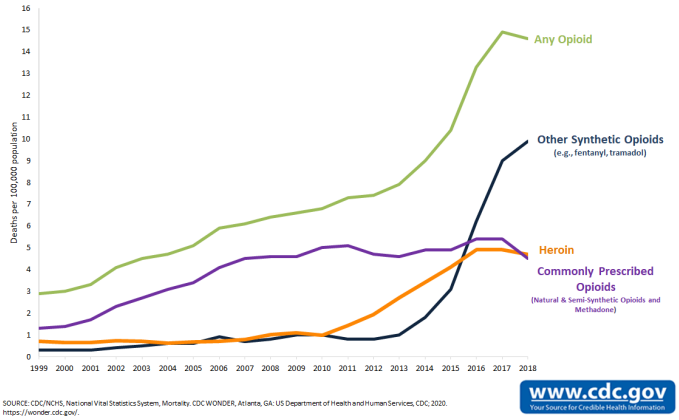


Fig. 6.1 Mortality Rate Involving Opioids from 1999 to 2018.

multifaceted approach, blockchain technology can help to tackle some issues [36] [149]. Blockchain is a decentralised ledger shared by all network participants and implemented with immutability using a cryptographic hash function (SHA-256) that is append-only with a time-stamped series of transactions called chain-connected blocks which serve as a database of past and present transactions [108]. This data structure allows provenance which includes a single place of origin for any transaction and because all transactions are unalterable, fraudulent activity can easily be tracked. This approach curbs prescription fraud activity by making it possible to determine the quantity of medication transferred, to whom the medicine was transferred, when it was transferred and the frequency of patient visits.

To offset these problems with the rise of the opioid epidemic, a blockchain-based system can set up a trusted network of hospitals and pharmacies to store opioid-related transactions including prescriptions, quantity prescribed, fulfillment, etc. in a secure and accountable manner. Also, the decentralised and distributed blockchain framework has the ability to work in a trustless manner with stakeholders by sharing an actual-time state-of-the-now database that remains in sync through consensus with an immutable spate of events. The resulting immutable ledger provides a record of drug transfers, ensures the supply chain's legitimacy and alerts authorities to potentially harmful or illegal distribution patterns.

The main focus of this research is the design and implementation of a secure prescription-tracking system between the provider, patient and the pharmacist on

blockchain using Hyperledger Fabric as the underlying permissioned blockchain technology. Moreover, the prescription updates can be sent to secure decentralized storage, IPFS, after secure cryptographic encryption. We aim to establish a new drug distribution blockchain platform where electronic prescriptions, medication dosage, and doctor and patient information are stored and exchanged efficiently across various hospital departments in a safe and approved network. Moreover,this patient centric approach employs smart contracts to facilitate medical transactions and consensus mechanisms to keep the system under control in the health data network. Fig. 6.2 shows an overview of the prescription process in the Healthchain in which the provider is facilitating a prescription and the pharmacist makes further uploads to IPFS after secure encryption. For a controlled prescription environment, features such as patient name or

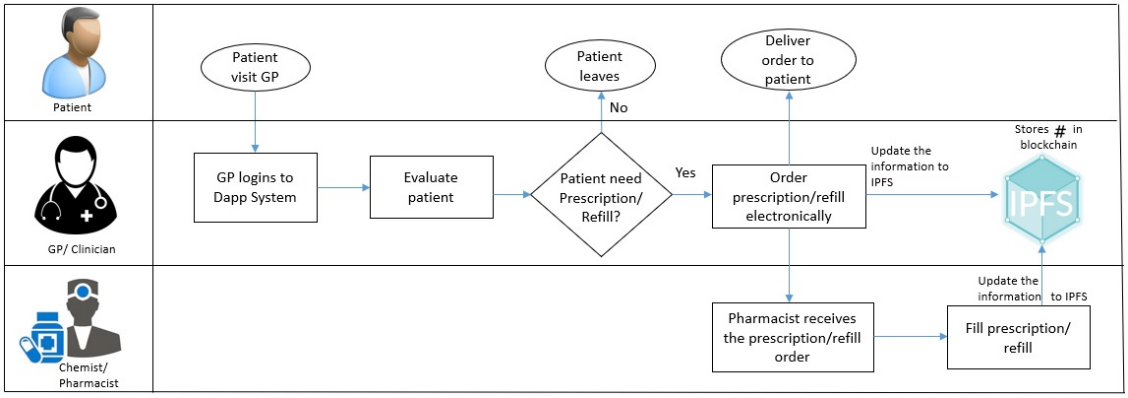


Fig. 6.2 Overview of Medical Prescription Process Flow in Healthchain.

ID, practitioners' name or ID, date of issue of the drug, drug name, drug strength, quantity prescribed, dosage form and number of refills authorized needs to be considered. The clinician prescription transactions, pharmacist's updates and record updates that invoke smart contracts create a unique hash and adds this to the healthchain.

This section also discusses some of the existing techniques proposed using the blockchain mechanism in healthcare management. MedRec is the first permissionless working healthcare application that employs smart contract functionality of Ethereum to represent medical records stored in the network's individual nodes [44]. Ancile and Medrec have scalability issues which can be resolved using IPFS through the secure offchain storage instead of the chains itself [11] [43] [17]. Furthermore, blockchain extends the existing data management system for personal health records (PHRs) to

integrate event-driven smart contracts to enable transactional services such as repeat prescriptions, scheduling appointments, and referral requests [80]. Sylim et.al. proposed a DApp-based smart contract system by employing Ethereum and Swarm as distributed file system for surveillance in the pharmaceutical supply chain system [137]. From the detailed studies conducted and investigated by Schneberk et.al. [137], it is of the utmost importance in the surveillance of the pharmaceutical drug supply chain management system to prevent prescription abuse and doctor shopping. Various security and privacy preserving solutions have been designed to protect the network against cyber-attacks [147] [148] [85]. Most of the existing solutions do not guarantee the vital requirements for Electronic Health Records (EHRs), such as data privacy, security, secure storage, efficient access control, scalability and interoperability. Our research work addresses most of the existing challenges by incorporating a novel encryption technique, access control rules and a smart contract functionality to demonstrate system feasibility.

The rest of the chapter is organised as follows: Section II presents the architecture of the proposed framework, Section III presents the implementation and simulation results and Section IV provides the summary.

6.2 Proposed System Architecture

An overview of the proposed system architecture is shown in Fig. 6.3. This represents the medical healthchain cycle with stakeholders such as doctor, patient and pharmacist and the blockchain that manages data related to drug, drug dose, and prescriptions. IPFS is the offchain decentralised database for the secure storage of all the health records for internal and external organisation. The hash generated by IPFS is stored in the blockchain and the state database CouchDB visualizes the internal blockchain structure. The doctor can access the patient's records with the patient's approval and the patient can also further share their health records with any authenticated doctors in the network. The permissions can be determined by the access control rules and smart contracts in the healthchain framework. The system developed includes reliable nodes for executing

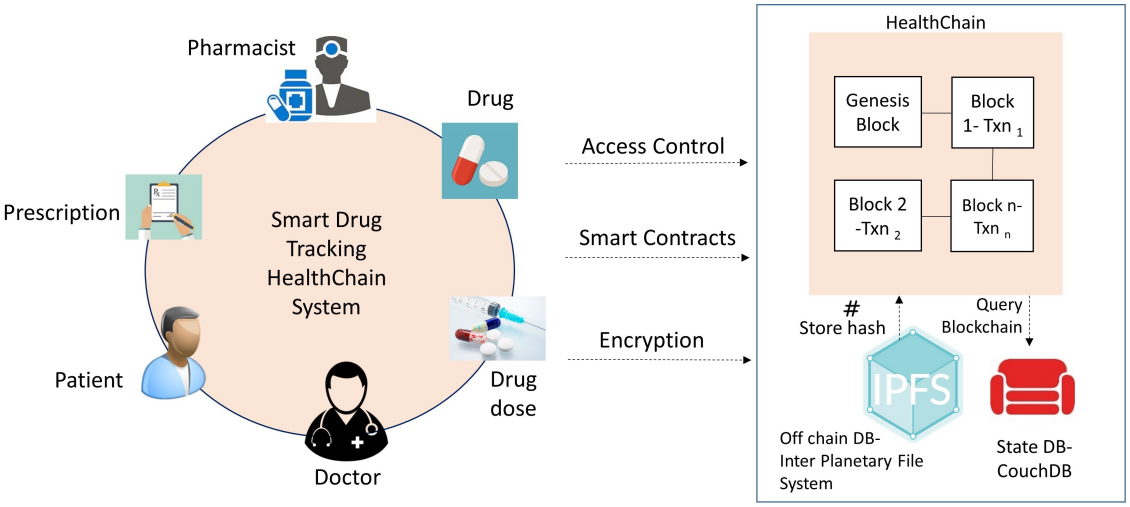


Fig. 6.3 System Architecture.

a consensus protocol for distributed ledger consistency. The doctor first evaluates the patient, prescribes the medication and drug dosage and provides other advice in the form of a computerised prescription. This prescription is then sent to the authenticated pharmacist to deliver the proper medication. The pharmacist checks the authenticity of the prescription, views the prescription, delivers the order and confirms the updates in IPFS. The pharmacist can only read the drug information related to the patient. The application developed is a patient-centric framework that employs smart contracts and distributed ledger as a middle-ware user service. The transaction request in the proposed system is submitted by the end user (i.e. doctor, pharmacist, receptionist or patient) via the application provided by the proposed blockchain network to access back-end services such as medical prescriptions, the profile management of stakeholders, patient appointments, EHRs, electronic pharmacy records (EPRs), pharmacy management, etc. The prescription component file defines how the prescription is managed, updated and shared with the chemist and a snippet of the component file is shown in Appendix B.4.

6.2.1 Transaction Flow in the Proposed Framework

This prototype is designed with a few stakeholders, namely doctor (clinician), patient, receptionist and pharmacist and builds a private healthchain framework. In this proposed framework, we define three entities viz patient, doctor and pharmacist for

interacting with the blockchain network. These entities communicate with the web application via Hyperledger Fabric SDK and the Composer rest server API. The assets and the transactions performed will be stored in the couchDB i.e. the state database and this work also proposes an off-chain database IPFS that can store diagnostic documents, such as huge-sized images or videos [7]. The work also proposes an efficient cryptographic algorithm for storing the data in IPFS. The prescription-based system works as shown in Fig. 6.4 and the steps are as follows:

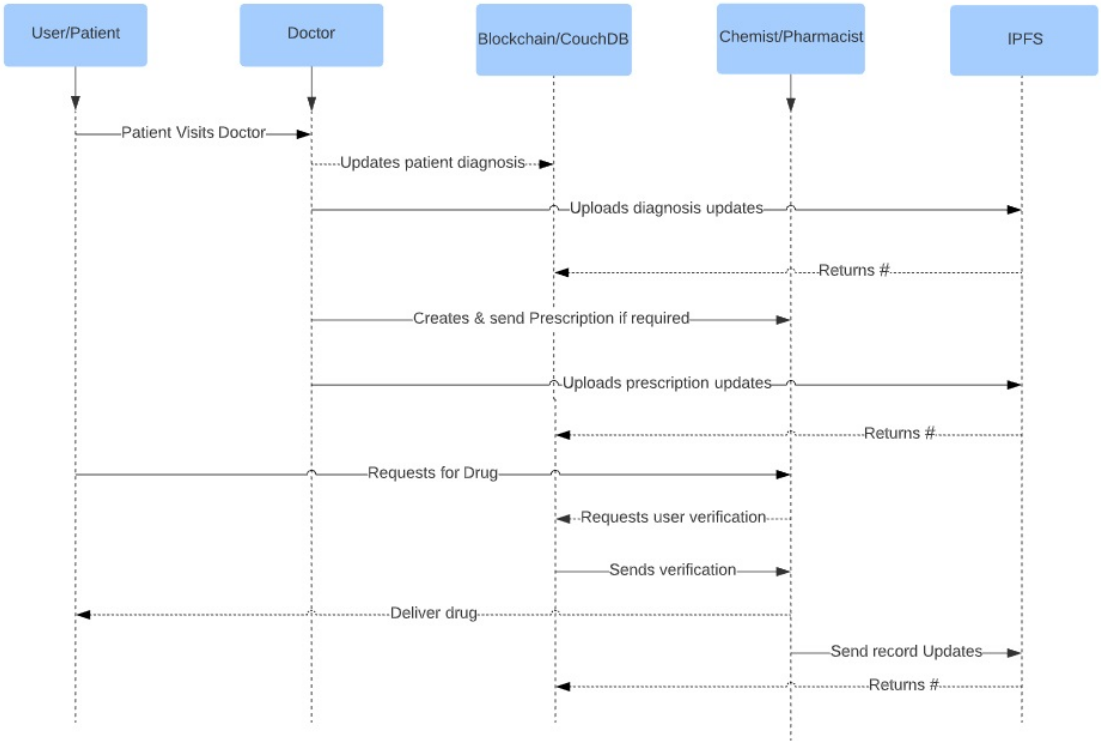


Fig. 6.4 Process Flow in the Proposed Framework.

1. The framework allows the patient to visit the authenticated doctor and the doctor updates the initial patient diagnosis in the blockchain. The doctor uploads the diagnosis updates to the IPFS which returns a hash value to the blockchain database.

2. The doctor provides the prescription after a careful examination if required, which is then added via a web application to the blockchain. The doctor can set the drug description, drug dose and even the drug expiry date to prevent this from being misused by the patient.

3. The patient requests the drug from the pharmacist. The pharmacist (chemist) verifies the user, checks the prescription validity and delivers the drug as a valid request else the pharmacist rejects the drug request.

4. The pharmacist confirms the drug transfer and send updates to IPFS which returns a hash value for that transaction to the blockchain, thus preserving data integrity.

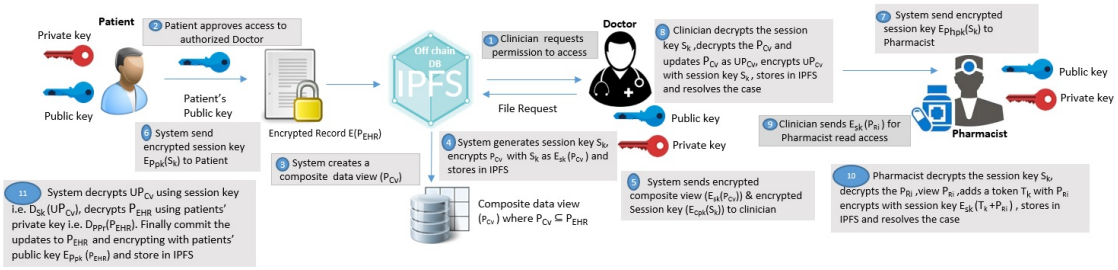


Fig. 6.5 Cryptographic Process in the Proposed Framework.

The working prototype is built on a permissioned blockchain called HealthChain, by combining three peer nodes to create decentralised web applications within a single organisation. This organisation has three peer nodes and an ordering node with a single public channel to register the participants in the network. A single channel is designed so that the Hyperledger Composer can communicate with the peers via the channel. Practical Byzantine fault tolerance (PBFT) [8] is the consensus protocol used in this blockchain-based healthcare platform. Mining nodes are known as peer nodes in which the anchor peer node is chosen in a round-robin fashion from the peers. The anchor peer receives all the transactions from the network participants and validates the transactions to create a block and broadcasts to all peer nodes. Each peer node Peer_i holds a copy of the ledger. The ledger can be queried via the Composer rest server.

There are four stakeholders in the healthchain network H_N with n participants for each stakeholder. The Fabric-Certificate Authority issues public key certificates to all n participants such as patient, clinician, receptionist and pharmacist. There is a key pair for each participant in which P_{Pk_i} and P_{Pr_i} are the public and private keys of patient P_i, C_{Pk_i} and C_{Pr_i} are the public and private keys of clinician C_i , R_{Pk_i} and R_{Pr_i} are the public and private keys of the receptionist R_i and Ph_{Pk_i} and Ph_{Pr_i} are the public and private keys of the pharmacist Ph_i respectively where i=1 to n. The scenario in Fig. 6.5 gives

a detailed explanation of how the clinician, patient and pharmacist interacts to manage drug tracking transactions in the Healthchain framework.

Table 6.1 Explanation of Notations.

Notations	**Definition**
H_N	Healthchain network
P_{EHR}	Patients' Health record
IPFS	InterPlanetary File System
P_{Cv}	Composite data view
S_k	Session Key
C_{Pk}	Public Key of Clinician
C_{Pr}	Private Key of Clinician
P_{Pk}	Patients' public key
P_{Pr}	Patients' private key
Ph_{Pk}	Public Key of Pharmacist
Ph_{Pr}	Private Key of Pharmacist
R_{Pk}	Public Key of Receptionist
R_{Pr}	Private Key of Receptionist
P_i	Patient
P_{ID}	Patient ID
C_i	Clinician
C_{ID}	Clinician ID
R_i	Receptionist
Ph_i	Pharmacist
Ph_{ID}	Pharmacist ID
PT_{ki}	Prescription Token
P_{Ri}	Prescription Report
UP_{Cv}	Updated Composite view
UP_{EHR}	Updated Health Record

The designed framework is a role-based model in which patients, physicians, chemists and receptionists can register and be authenticated via a client application using user credentials such as email address and password. Patients can provide appropriate read, write and deny access for EHRs to stakeholders in the network. Patients can book a doctor appointment by themselves or via the receptionist. Permissioned doctors can create medical records in the network that invokes smart contracts to commit the transaction in the network. There are several smart contracts defined in this framework for the transactions viz CreateMedicalRecord, UpdateMedicalRecord, GrantPharmacistAccess, RevokePharmacistAccess etc. All the

transactions are distributed across the healthchain network in which only authenticated stakeholders can access documents which are allowed access. Each node in the framework holds a copy of the ledger and all the committed transactions are distributed across the nodes creating a decentralised network. Fig. 6.5 illustrates the cryptographic process in the proposed framework. A detailed explanation of the cryptographic process is explained with the proposed algorithms 1, 2 and 3.

6.2.2 Proposed Algorithms

This framework has four stakeholders, doctor, patient, pharmacist and receptionist with n users for each participant. Algorithm 1 illustrates the patient working in the Healthchain network, Algorithm 2 illustrates clinician working in the Healthchain network and Algorithm 3 illustrates the pharmacist working in the Healthchain network. Table 6.1 explains the notations used in the algorithm. The patient has read access to their own health records and can provide read, write, revoke and deny access permissions to the authenticated stakeholders in the network. If P_{EHR_i} is not in the network, then the patient provides access to the clinician to create P_{EHR_i}. For an existing record upon the clinician's request, the system creates a composite view P_{Cv_i} of the patient record P_{EHR_i}, alternately sharing the whole medical record of the patient as shown in step 15 of Algorithm 1. Composite view P_{Cv_i} is the attribute set of the stored medical record P_{EHR_i} that the system creates on permissioned user request without sharing the complete patient record. In other words P_{Cv_i} is a subset of P_{EHR_i} as shown in equation 6.1.

$$P_{Cv_i} \subseteq P_{EHR_i} \tag{6.1}$$

$$P_{Cv_i} = (D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i}))) \tag{6.2}$$

$$P_{EHR_i} = [(D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i}))) + (E_{P_{Pk_i}}(U_{P_{Cv_i}}) + (PT_{ki}))] \tag{6.3}$$

Algorithm 1 :Algorithm on Patient working

Input: P_{ID} and P_{Pk}
Output: Get Access to Patient ledger transactions $P_L \in H_N$
Initialisation : P_L should be a valid node and can Read, Revoke, Grant or Deny EHR records

procedure Patient (P_{ID})
while (True) do
if ($P_{ID} \in H_N$) **then**
  if ($P_{EHR_i} \notin H_N$) **then**
    create_records(P_{ID}, P_{EHR_i}, H_N)
  else
    read_records(P_{ID}, P_{EHR_i}, C_{ID}, H_N)
  end if
else
  P_{ID} is invalid
end if
if visit (P_{ID} ,C_{ID}, H_N) **then**
  P_{EHR_i} = Medical_record (P_{ID})
  if ($P_{EHR_i} \in P_L(H_N)$) **then**
    $P_{CV_i} \leftarrow \int_{i=1}^{n} (D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i})))$
    Grant_records(P_{CV_i}, C_{ID}, S_k, H_N) where $P_{Cv_i} \subseteq P_{EHR_i}$
    $C_i \leftarrow E_{C_{Pk_i}}(S_k)$
    $C_i \leftarrow E_{S_k}(P_{Cv_i})$
    Algorithm2 ()
  else
    (C_{ID}) $\leftarrow$ NOTIFY ("Medical records does not exist")
  end if
  if (UP_{Cv_i}) **then**
    $P_{EHR_i} \leftarrow [(D_{P_{Pr_i}}(E_{P_{Pk_i}}(P_{EHR_i}))) + E_{P_{Pk_i}}(UP_{Cv_i}) + E_{P_{Pk_i}}(PT_{ki})]$
  end if
  if ($P_{EHR_i} \in C_{ID}$, treatment completed (P_{ID})) **then**
    end session (S_k, P_{EHR_i}, C_{ID})
  else
    (C_{ID}) $\leftarrow$ NOTIFY ("voluntary revoke P_{EHR_i}")
    Revoke_records(P_{EHR_i}, C_{ID}, H_N)
  end if
else
  Not visit
end if
end while
end procedure

The system then creates a common session key between the clinician, patient and pharmacist for a specific session. The system sends the encrypted session key $E_{P_{pk_i}}(S_k)$ and composite view $E_{S_k}(P_{Cv_i})$ to the clinician. The clinician decrypts the session key with the private key and decrypts the composite view with the session key and if there are any updates, updates P_{Cv_i} as UP_{Cv_i}, resolves the case, encrypts it with the session key

Algorithm 2 :Algorithm on Clinician working

Input: C_{ID} and C_{Pk}
Output: Get Access to Clinician ledger transactions $C_L \in H_N$
Initialisation : C_L should be a valid node and can Read or Write EHR records permissioned by the patient

1: **procedure** Clinician (C_{ID})
2: **while** (True) do
3: **if** ($C_{ID} \in H_N$) **then**
4: **if** (Granted P_{EHR_i}) **then**
5: Read_records(C_{ID}, P_{EHR_i}, H_N)
6: Update_records (C_{ID}, UP_{Cv_i}, H_N)
7: $C_i \leftarrow D_{C_{Pr_i}}(S_k)$
8: $C_i \leftarrow D_{S_k}(P_{Cv_i})$
9: $P_{Cv_i} \rightarrow (UP_{Cv_i})$
10: IPFS $\leftarrow E_{S_k}$ (UP_{Cv_i})
11: **end if**
12: Pharmacist $\leftarrow E_{S_k}(P_{Ri})$
13: **Algorithm3 ()**
14: **else**
15: C_{ID} is invalid
16: **end if**
17: **end while**
18: **end procedure**

Algorithm 3 :Algorithm on Pharmacist working

Input: Ph_{ID} and Ph_{Pk}
Output: Update Prescription confirmation to Patient Ledger transactions $P_L \in H_N$
Initialisation : Ph_{ID} should be valid and can Read Prescription records permissioned by the Clinician

1: **procedure** Pharmacist (Ph_{ID})
2: **while** (True) do
3: **if** ($Ph_{ID} \in H_N$) **then**
4: **if** (Granted P_{Ri}) **then**
5: Read_records(Ph_{ID}, P_{Ri}, H_N)
6: Pharmacist $\rightarrow$ (PT_{ki})
7: IPFS $\leftarrow E_{S_k}$ (PT_{ki})
8: **end if**
9: **Algorithm1 ()**
10: **else**
11: Ph_{ID} is invalid
12: **end if**
13: **end while**
14: **end procedure**

and uploads UP_{Cv_i} to IPFS as $E_{S_k}(UP_{Cv_i})$ as shown in step 10 in Algorithm 2. If there are any prescriptions, the clinician sends a prescription update P_{Ri} to the pharmacist as shown in step 12 of Algorithm 2. The pharmacist reads the prescription and delivers the drug if the request is valid or else denies the request. Moreover, the pharmacist sends a

prescription token PT_{ki} and updates $E_{S_k}(PT_{ki})$ on placing the prescription as shown in step 7 of Algorithm 3 and resolves the case. Finally, the system commits the updates to the original record and encrypts the original record P_{EHR_i} before uploading it to IPFS as shown in step 24 in Algorithm 1 and equation 6.3.

6.2.3 Access Permission Rules

Fig. 6.6 illustrates the access control permission rules used in the proposed framework. By defining the access control language (ACL) rules, we can decide which users or roles in the domain model are allowed to build, read, update or delete resource components in the blockchain business network. From Fig. 6.6, it is evident that the chemist has read access to EHR only if the subject ID matches with the resource ID of the patient.

The algorithm initially verifies the access permission rules for granting or denying access to the health records. Access management mechanisms are designed to preserve the security and privacy of the patients' health records from unauthenticated access. Algorithm 2 explains the process of clinician record creation and updates in the blockchain network. When an access request is made by the user, the algorithm verifies the permission rules for access control that determine the user's access rights to the owner's defined EHR resource. These access rules are stored in the blockchain and sent via a transaction called the Business Network Archive Transaction to the blockchain channel. In this approach, the rules comprise the description, operation and condition specifying the subject ID to which the access control policy grants access. Also, the conditions specify the sets of values authorized for the subject, resource, action type and transaction attributes for access to be granted. This framework designs the rule to correctly change these requirements as they transfer access rights to other authenticated users prior to submitting it to the healthchain. The actors in this scenario are resource owner P, Resource EHR_i and several subjects such as C_i, Ph_i and R_i in the healthchain framework. The clinician C_i or any user can only read, write, modify or update access to the health records in accordance with the access control permissions. From the Fig. 6.6 it is clear that if the rules match with the subject request and only if the subject is a

- rule DoctorCanReadPatient
  description:"Allow doctor read access to all granted patients"
  participant(p):"org.ehr.healthchain.Doctor"
  operation:READ
  resource(r):"org.ehr.healthchain.Patient"
  condition(r.authorized && r.authorized.indexOf(r.getIdentifier())>-1)
  action:ALLOW
- rule DoctorCanUpdateEHR
  description:"Allow doctor update access to all granted patients"
  participant(p):"org.ehr.healthchain.Doctor"
  operation:CREATE,UPDATE
  resource(r):"org.ehr.healthchain.Patient"
  transaction(tx):"org.ehr.healthchain.UpdateRecord"
  condition(r.authorized && r.authorized.indexOf(p.getIdentifier())>-1)
  action:ALLOW
- rule ChemistCanReadEHR
  description:"Allow chemist read access to all granted patient records"
  participant(p):"org.ehr.healthchain.Chemist"
  operation:READ
  resource(r):"org.ehr.healthchain.Medical_Record"
  condition(ph.ChemistId==r.PatientId)
  action:ALLOW
- rule DoctorCanUpdatePatientPrescriptionDose
  description:"Allow doctor update access to all granted patients"
  participant(p):"org.ehr.healthchain.Doctor"
  operation:READ,CREATE,UPDATE
  resource(r):"org.ehr.healthchain.UpdateMedical_Record"
  condition(r.authorized && r.authorized.indexOf(p.getIdentifier())>-1)
  action:ALLOW

Fig. 6.6 Access Permission Rules in the Proposed Framework.

permissioned stakeholder, permissions such as read, write access are allowed or otherwise access will be denied.

6.3 Prototype Implementation and Results

We initially used a private Hyperledger Fabric blockchain to implement our proposed Healthchain platform for a single organisation containing three peer nodes which hold a copy of the ledger with a shared ordering service in a Linux environment where smart contracts are deployed for each transaction in the healthchain, CouchDB is the state

internal database, the IPFS storage system is utilized for storing huge data and the network entities are developed to create the Healthchain system. The simulation is conducted in a virtual machine environment and the PC has the configurations, as shown in Table 4.3. Fig. 6.7 illustrates the process of creating the prescription in the doctor's profile in the Healthchain framework. The asset has been created in the doctor's profile with unique attributes such as recordId, patientId, doctorId, chemistId, drugdescription, quantity prescribed and appropriate medical files that need to be seen by the chemist for the drug prescription. Fig. 6.8 illustrates the EHR prescription created in the doctor's

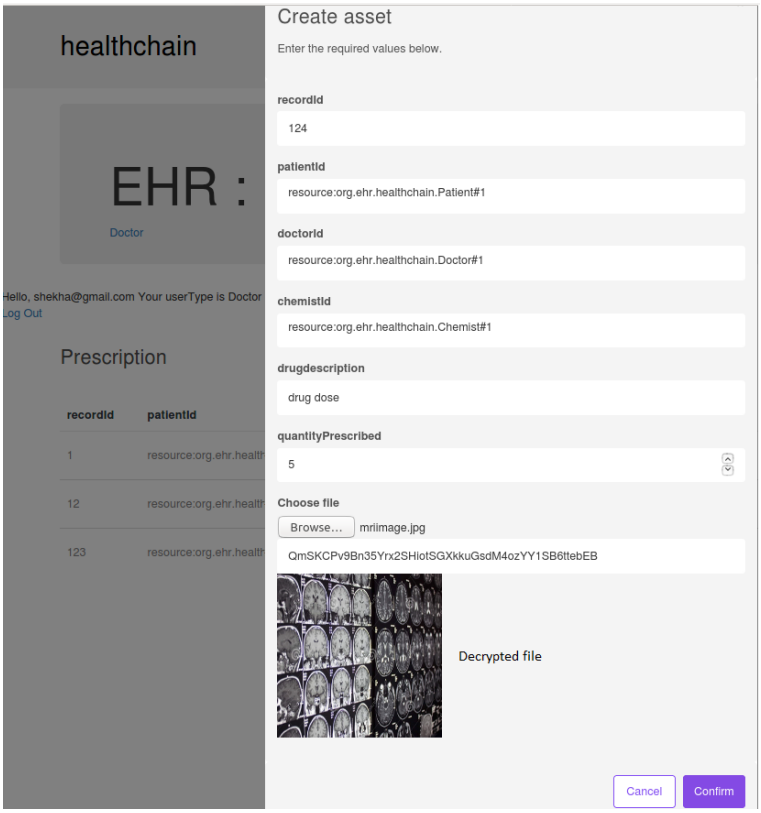


Fig. 6.7 Illustration of EHR Prescription Creation in the Doctor's Profile in the Proposed Healthchain.

profile in the framework. The prescription created is reflected in the chemist profile with the required attributes as shown in Fig. 6.9. The prescription from the doctor to the chemist invokes the smart contracts to reflect the transaction information updates in the ledger across the Healthchain network nodes. Fig. 6.10 illustrates to process of querying the records via the Composer Rest Server API in the proposed system

Furthermore, this system also allows a patient to view the medical records and also

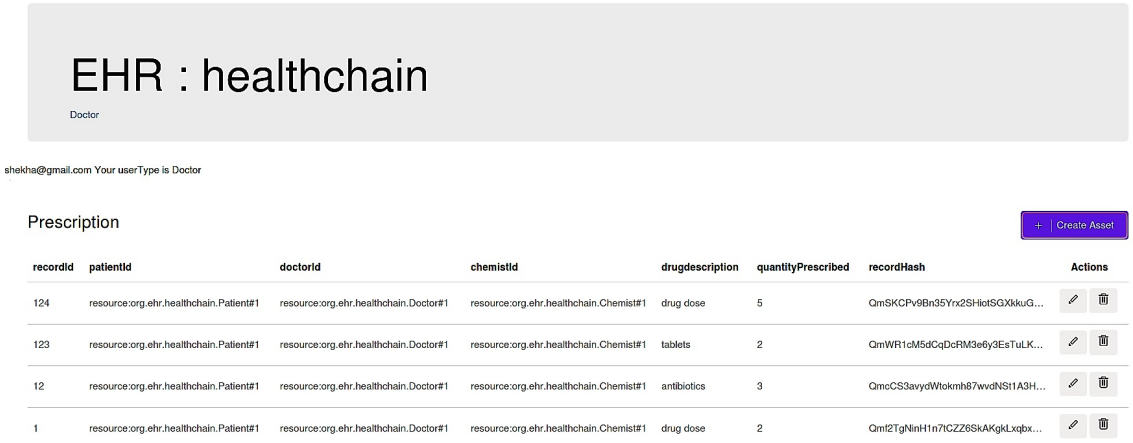


Fig. 6.8 Illustration of EHR Prescription in the Doctor's Profile in the Proposed Healthchain.

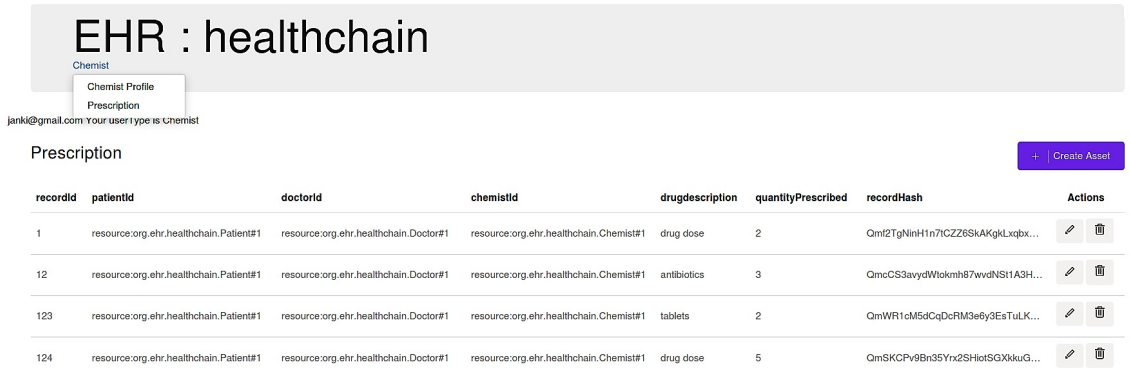


Fig. 6.9 Illustration of EHR Prescription in the Chemist's Profile in the Healthchain.

keeps a provenance history of the medical records, as shown in Fig. 6.11. The provenance history of the patients shows all the patient details or transaction details at every stage of the user in the network. Fig. 6.11 includes all the details, recordId, doctorId, record description, recordhash, encounter time and location. This makes it easy to track the record details that serves as a history wallet in the system.

6.3.1 Case Study and Framework Functionality

- Case I : Efficient Creation of Prescription
- Case II : Effective Security, Access Permissions and Scalability
- Case II : Efficient Provenance Management

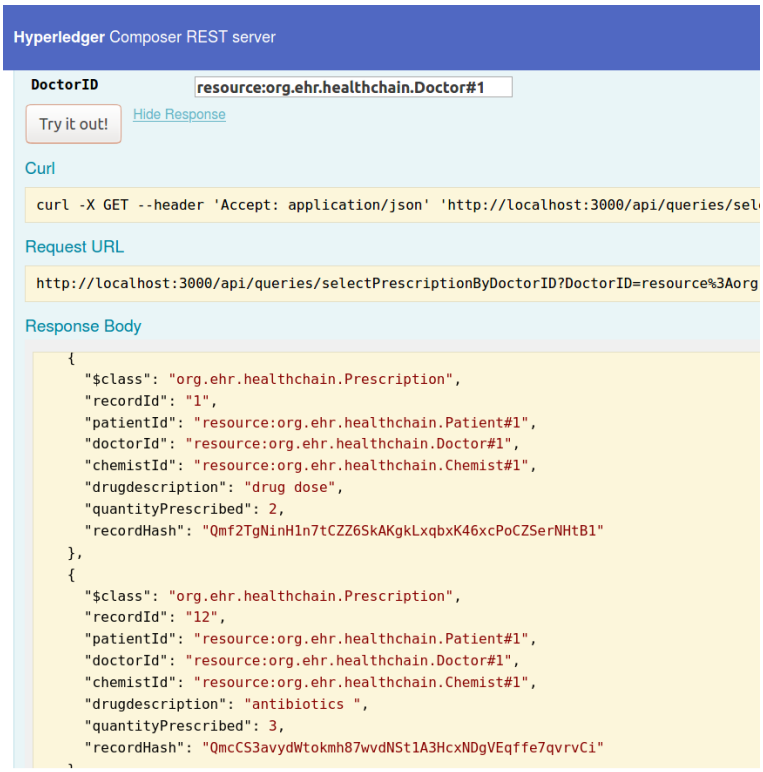
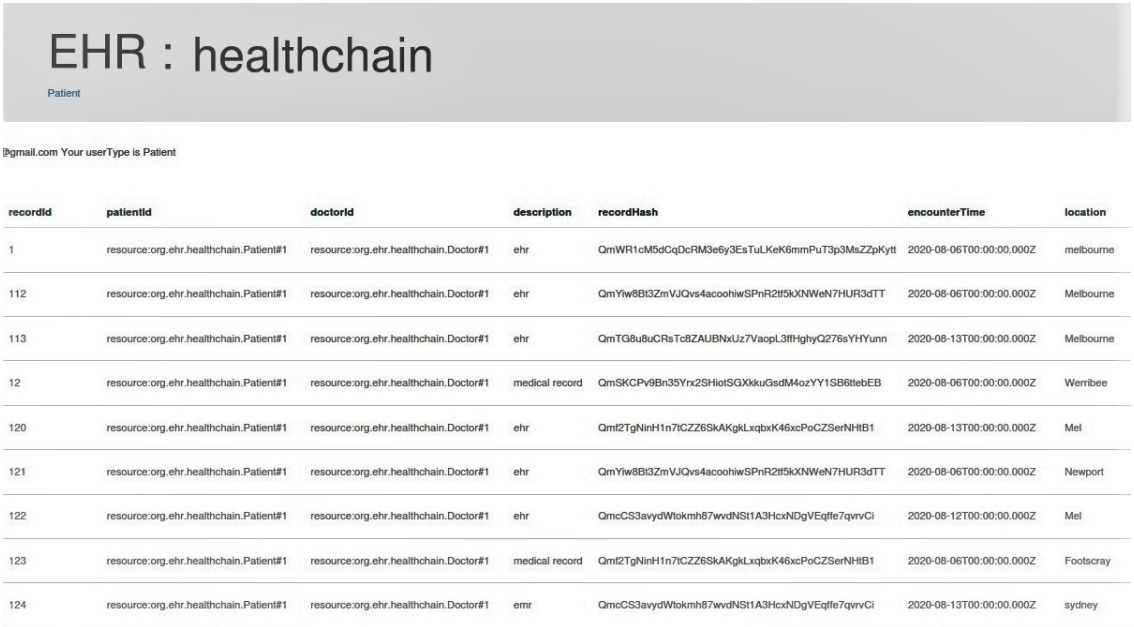


Fig. 6.10 Illustration of Querying the Health Records in the Proposed Healthchain.



recordId	patientId	doctorId	description	recordHash	encounterTime	location
1	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	QmWR1cM5dCqDcRM3e6y3EsTuLKeK6mmPuT3p3MsZZpKytt	2020-08-06T00:00:00.000Z	melbourne
112	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	QmYiw8Bt3ZmVJQvs4acoohiwSPnR2tf5kXNWeN7HUR3dTT	2020-08-06T00:00:00.000Z	Melbourne
113	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	QmTG8u8uCRsTc8ZAUBNxUz7VaopL3ffHghyQ276sYHYunn	2020-08-13T00:00:00.000Z	Melbourne
12	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	medical record	QmSKCPv9Bn35Yrx2SHiotSGXkkuGsdM4ozYY1SB6ttebEB	2020-08-06T00:00:00.000Z	Werribee
120	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	Qmf2TgNinH1n7tCZZ6SkAKgkLxqbxK46xcPoCZSerNHtB1	2020-08-13T00:00:00.000Z	Mel
121	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	QmYiw8Bt3ZmVJQvs4acoohiwSPnR2tf5kXNWeN7HUR3dTT	2020-08-06T00:00:00.000Z	Newport
122	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	ehr	QmcCS3avydWtokmh87wvdNSt1A3HcxNDgVEqffe7qvrvCi	2020-08-12T00:00:00.000Z	Mel
123	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	medical record	Qmf2TgNinH1n7tCZZ6SkAKgkLxqbxK46xcPoCZSerNHtB1	2020-08-06T00:00:00.000Z	Footscray
124	resource:org.ehr.healthchain.Patient#1	resource:org.ehr.healthchain.Doctor#1	emr	QmcCS3avydWtokmh87wvdNSt1A3HcxNDgVEqffe7qvrvCi	2020-08-13T00:00:00.000Z	sydney

Fig. 6.11 Illustration of Provenance History of Patient Health Records in the Proposed Healthchain.

Efficient Creation of Prescription- Case I : The efficient creation of a health record prescription in the clinician's profile is tested against a few cases listed in Fig.6.12. The first test case validates if a clinician can successfully create a prescription in the profile

and upload the results in IPFS. The implementation results in Fig. 6.8 show that the doctor authenticated by the patient can have write access for the medical records and also upload encrypted records into IPFS. The second case verifies if the authenticated doctor has read access permission and is successfully verified as the doctor who has been authenticated by the patient for a particular session. Furthermore, it checks that a patient can view the prescription records created by the clinician and Fig. 6.11 illustrates the provenance history of the medical records. Moreover, the system is tested whether a prescription record can be uniquely identified or not and is successful as every prescription record created is uniquely related to a record ID, doctor ID, chemist ID and patient ID. Additionally, the system is checked to see whether an encrypted record can be effectively retrieved after decryption and is successful as shown in the decrypted file component in Fig. 6.7. The outcome is successful as the updated record can be encrypted with the specialist doctor's session key for storing in IPFS and the updated record can be decrypted using the patient's session key at the patient side.

S.No	Test case	Description	Outcome
1	Verify if a doctor can create prescription in the profile	The doctor authenticated by the patient can have write access for the record	Passed
2	Verify if a doctor can view the prescription records on permission	The specialist doctor authenticated by the patient can have read access for the records	Passed
3	Verify if a patient can view the prescription	The patient can see the provenance history of the patient records	Passed
4	Verify if a prescription created is uniquely identified	Each prescription is uniquely associated with a recordId, patient Id, doctor id and chemistId	Passed
5	Verify if an encrypted record can be effectively retrieved	The updated medical record can be encrypted with doctor's session key for storing in IPFS and updated record can be decrypted by using patients' session key at the patient side	Passed

Fig. 6.12 Efficient Creation of Prescription in Healthchain.

Effective Security, Access Permissions and Scalability- Case II : The degree of security, access control and scalability in the healthchain has been verified against a few test cases, as shown in Fig 6.13. By checking the first case, security is maintained and is proven successful as the clinician sends the encrypted record for the pharmacist to access for a definite session. Additionally, the system is tested to check whether a patient can grant read access, revoke access and give access permissions to the health records to the stakeholders and is successful in preserving data privacy. Furthermore,

the system was also tested to see whether the pharmacist can upload the updated prescription details to IPFS and the result is successful as the pharmacist utilizes session key encryption before the session expires. This research further contributes to data scalability by enabling records of size 1000 MB to be uploaded at a time to IPFS and is successful which improves the scalability of the system.

S.No	Test case	Description	Outcome
1	Verify if a clinician can provide secure prescription to the pharmacist (chemist) in healthchain	Clinician sends encrypted prescription after session key encryption for pharmacist access for a definite session	Passed
2	Verify if a patient can grant read access ,revoke access, and provide access permission of prescription with the pharmacist	The patient has all the permissions to grant or revoke access from other user types to maintain privacy and patient has the permission to provide read, write, and deny access permission according to the role type	Passed
3	Verify if the pharmacist can upload the updated details in IPFS	The pharmacist encrypts the prescription update with session key and upload to IPFS before the session expires	Passed
4	Verify if the clinician and pharmacist can upload huge files in IPFS	Upload records with size > 1000 MB	Passed
5	Verify if the clinician and pharmacist can upload huge files with different extension in IPFS	Upload text and image files. files with extension such as video, audio files can be tried in a later stage	Future Work

Fig. 6.13 Security, Access Permissions and Scalability in Healthchain.

Efficient Provenance Management - Case III The healthchain system tested the provenance management using three test cases. The first case tested whether the patient can view all the medical records and is successful as the patient can view the provenance history of the records as shown in Fig. 6.11. The system tested the clinician's and pharmacist's access to prescription history and is found successful, as shown in Fig. 6.8 and Fig. 6.9.

S.No	Test case	Description	Outcome
1	Verify if the patient can view all the record details	The patient can see the provenance history of all the medical records	Passed
2	Verify if the doctor can view all the prescription records	The doctor can see the prescription history of the records	Passed
3	Verify if the pharmacist can view all the prescription records	The pharmacist can see the prescription details and have read access	Passed

Fig. 6.14 Provenance Management in Healthchain.

6.3.2 Performance Analysis

Fig. 6.15 shows the scalability of storing health records in IPFS. IPFS stores the records in different nodes if the size is greater than a particular threshold (greater than 256KB). For this research, the records will be stored in a way to ensure it is cryptographically protected after encryption with the specified encryption algorithm. Fig. 6.15 shows the record uploading and downloading time of five concurrent users in the healthchain network. Considering the machine configuration, the system takes an average of 60 seconds to upload the data to IPFS and 80 seconds to download a 100MB data from IPFS. This research results in improved security with the proposed encryption, improved privacy with the defined access control rules, enhanced integrity with the proposed blockchain framework and improved scalability with the introduction of IPFS for decentralised data storage.

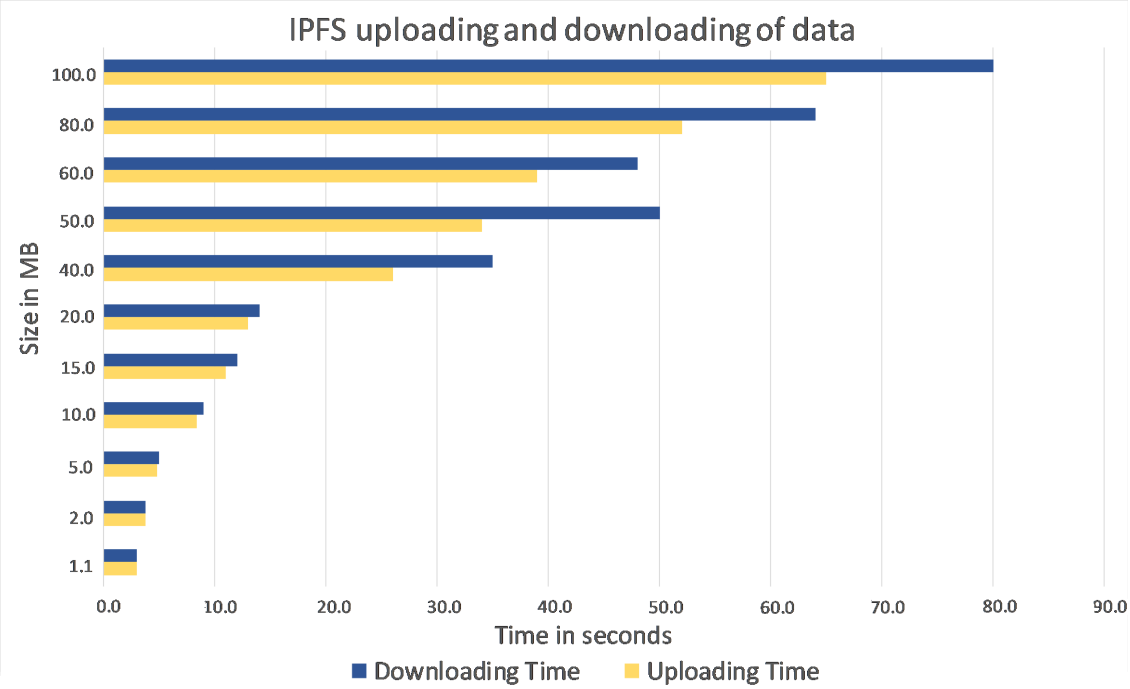


Fig. 6.15 Scalability in IPFS.

Several experiments were conducted to test transaction latency of the proposed framework. Transaction latency is the time needed to commit the transaction and is available across the network nodes. The transactions used in this experiment are: (a) Create Medical Records (b) Update Medical Records (c) Update Ownership (g)

Prescription to Pharmacist. Fig. 6.16 shows the transaction latency of the clinician to the pharmacist prescription update in the network among the peer nodes. If there are n number of nodes in the blockchain network in which T_{L_n} is the transaction latency and the confirmation time is T_{C_n} in the network nodes and the transaction submit time in seconds is T_{S_n} then;

$$T_{L_n} = T_{C_n} - T_{S_n} \tag{6.4}$$

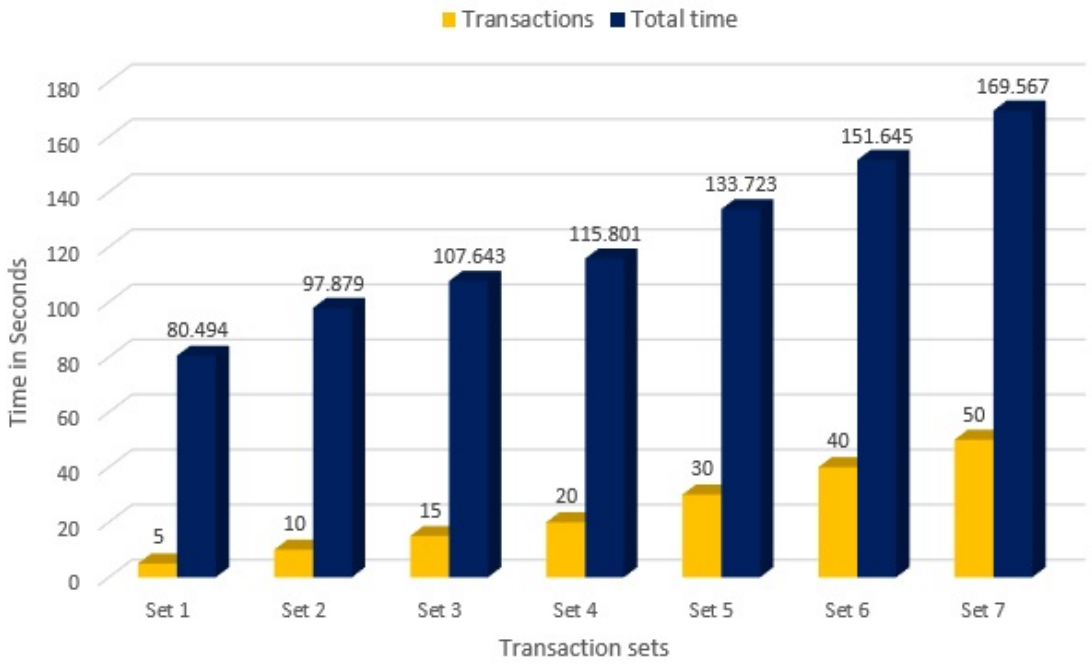


Fig. 6.16 Transaction Latency.

The experiments are executed in three peer nodes with seven sets of transaction commit to the network ledger in transaction sets of varying size of 5, 10, 15, 20, 30, 40 and 50 as shown in Fig. 4.19. Considering the machine configuration, it can be seen that the first 5 sets of transactions take an average of 80 seconds to commit, the second 10 sets of transactions take an average of 97 seconds to commit and the last set of 50 transactions take an average of 160 seconds to commit across the network. It is therefore apparent that with an increase in peers and an increase in the number of transactions, the time required to execute transactions increases.

6.3.3 Comparison of Framework with Existing Techniques

This section conducts a comparative analysis of the smart drug tracking healthchain system with the existing e-prescription blockchain-based systems in terms of main privacy preserving requirements viz data security, patient privacy, data integrity, data privacy, consensus, provenance, confidentiality and scalability. The proposed framework is compared against the existing blockchain-based implementations such as [137], [73] and [141]. From the table 6.2, it is evident that the proposed system addresses the shortcomings of the existing systems in terms of data security, data integrity, privacy, scalability and data provenance. This section also describes how the proposed framework satisfies the privacy preserving requirements.

Table 6.2 Comparative Analysis.

Scheme	Data Integrity	Data Privacy	Data Security	Confidentiality	Scalability	Provenance
Supply chain[137]	✗	✓	✓	✓	✗	✗
Smart contract healthcare system[73]	✓	✓	✓	✓	✗	✗
PDMP[141]	✓	✗	✓	✓	✗	✗
Proposed Framework (Drug Tracking System-Healthchain)	✓	✓	✓	✓	✓	✓

Data Integrity : Data integrity is maintained as the data is stored as hash values in each block and trust in this blockchain framework is based on consensus, digital signature and the designed cryptographic algorithm despite relying on a third-party provider. Since all the blocks are linked, any modification in the original data results in a change in its hash value and it is computationally difficult to tamper with the ledger, hence that the immutability of the medical records are explicitly guaranteed. In addition, IPFS stores the data after performing a special cryptographic encryption technique and stores the data in multiple nodes if the size of the data is greater than a defined threshold.

Data Privacy : The smart drug tracking framework ensures fine-grained access control by integrating role, rule and attribute-based access control permission rules for any data request. Secondly, unauthenticated data access is restricted since the blockchain only stores the hash value of the encrypted medical record. Thirdly, if the data requester attributes do not meet the access policy embedded in the network archive

file, it is also impossible to acquire any real medical record data from the blockchain public information.

Data Security : This framework utilizes a patient-centric approach which provides authenticated access permissioned by the patient and guarantees data security. Moreoever, the smart contract functionality for every transaction combined with blockchain solutions embraces high-level encryption and ensures patient confidentiality. In addition, the data stored on IPFS is encrypted using a special cryptographic algorithm to establish robust blockchain data solutions.

Confidentiality : In this framework, every health record of the patient is stored in the IPFS after encrypting it with the patient's public key and allows only the permissioned user to access the record for a particular session. Since the framework is a patient-centric approach in which the patient has complete control to provide access permissions to the stakeholders, except in emergency situations, the confidential nature of the health data is preserved.

Scalability : The proposed scheme employs IPFS as the decentralised storage for health records and stores the encrypted data in different nodes, thereby resolving the scalability issues in the existing techniques. The scalability of the proposed system is demonstrated and it is proven that the system is capable of processing large datasets with low latency, as shown in Fig. 6.15.

Provenance : The provenance of the record defines the recorded history of the actors, their operations, procedures and communications relevant to the development and modification of the data. This framework supports provenance history by storing the users' metadata in the Healthchain system.

6.4 Summary

In this research work, a permissioned blockchain framework has been implemented for secure drug prescription tracking between stakeholders in healthcare utilizing Hyperledger Fabric and Hyperledger Composer. This work created smart contracts for various medical work-flows, and then data access permissions are managed by the patient in the healthcare ecosystem. Moreover, this research proposes an efficient cryptographic mechanism for securing data storage and providing efficient access control between stakeholders viz patients, doctors, pharmacists and other participants via encryption techniques and access control mechanisms. A working prototype based on Hyperledger Fabric and the IPFS is made to illustrate the system's viability and consequently, the framework is proven successful as a reliable health data network. With healthcare data growing each year, we look forward to improving this prototype with robust scalability simulations and comparing it with other blockchain architectures in a test bed arena that invites more interest in future research work.

Chapter 7

Conclusion and Future Work

This chapter concludes the thesis by encapsulating the major contributions of this research study, including the limitations of the research, future directions and its wider research impact in the field of cyber security.

7.1 Summary of Contributions

Electronic health records in healthcare have experienced problems with privacy breaches and unauthenticated record access in recent years, the prime one related to the privacy and security of medical data. Since the right to privacy is fundamental, there is an enormous need to protect data from possible breaches to ensure patient confidentiality. The misuse of patient health data may harm patients and undermine the quality of health care. Since most of the data is sensitive and strictly confidential, security is a major concern. Patient privacy is paramount for healthcare organisations, including hospitals, medical centres, independent physician groups and insurance providers. In Australia, millions of healthcare documents are sent across the country. As the data is stored in third-party cloud servers where the user does not have direct control, the need to provide privacy and security increases. The records of patients with chronic conditions and sensitive information on patients' needs to be securely shared

and accessed between health care providers. This research also focused on identifying the most appropriate method to share private information between multiple providers in the patient's care team and the patient and their family or carers. This research ensures that both the patient's privacy and the data are securely maintained.

Since e-health data contains various sensitive and confidential information ranging from patient data to financial information, such as social security number, credit card details, data leakage not only throws open the patients' sensitive information and has the potential to cause financial losses, it also infringes the most fundamental right of a citizen in any country i.e. the right to privacy. Certain privacy-preserving mechanisms exist in the literature but are not adequate to ensure foolproof security in the e-health cloud. The main issue affecting health records in cloud servers is internal attacks by those who have authorized credentials within an organization to access data, where the database administrator or the key manager is the attacker, which is significantly worse than external attacks. Another major threat is the openness of data to cloud providers which poses the dangers of data threat or misuse. This scenario motivated this thesis to devise a new mechanism which offers better safety and security measures in the e-healthcare infrastructure. Most of the aforementioned problems are resolved by our blockchain technology named Healthchain in the e-health environment which provides efficient scalability for electronic health records and secure record sharing in the e-health environment that offsets the shortcomings in the existing system and ensures a better infrastructure in providing privacy and security for e-health data.

Also the increase in cyber-attacks adversely impacts the health care sector at an alarming rate. As the healthcare sector continues to offer life-critical services while working to improve treatment and patient care with new technologies, criminals and cyber threat actors look to exploit the vulnerabilities that are coupled with these changes. These issues range from malware that compromises the integrity of systems and the privacy of patients to distributed denial of service (DDoS) attacks that disrupt a facility's ability to provide patient care. For healthcare, cyber-attacks like Ransomwares can have ramifications beyond financial loss and breaches of privacy. This work also aimed for protecting the privacy of patients by strengthening security to prevent possible

breaches of data. To address all the existing issues, the overall aim of the research was to develop a new task-based framework for data sharing on Electronic Health Data (EHD) database federations while protecting data against both outsider and insider attacks, and providing visualised, dynamic support to medical staff and government resource planners and policymakers. The individual objectives of this study are as follows:

• *Empower medical research .i.e. to establish a system to see how the approved blockchain applications may be useful to manage the privacy and protection of medical information when health data are shared or accessed by stakeholders.*

• *To introduce a secure storage system and also devise a cryptographic mechanism to provide efficient and secure data storage.*

• *To develop a framework to Improve the privacy protection against insider attacks and outsider attacks.*

The research objectives led to the following contributions and outcomes:-

This thesis offers cost-effective and resilient blockchain deployment for EHR systems to enhance auditability and privacy. The proposed blockchain framework Healthchain, has been successfully implemented on Hyperledger Fabric (**Chapter 4**). The proposed framework builds chaincode implementations called smart contracts for the proper functioning of the system upon transaction execution. Moreoever, in the implementation of the framework, unlike a conventional transaction flow, instead of assigning an ordering authority to construct the block in the framework, the process choses peer pairs that support a more computationally intensive job. In addition, this smart contract implementation places EHR transactions as immutable hash values in the Healthchain network and the access control permission rules packaged in the business network definition (.bna file) safeguards against health records access by malicious users. Also, the special cryptographic encryption approach used in this prototype for secure storage protects patient privacy from harmful attacks.

The individual objectives have been met in this research work in which a permissioned blockchain framework was implemented for secure data storage and

access to electronic health records utilizing Hyperledger Fabric and Hyperledger Composer (**Chapter 4**). Since the healthchain is tamper-resistant, the system is tamper-proof and can preserve the data privacy, security and integrity of healthcare records. Moreover, no incentive mechanisms for blockchain mining are included that demonstrates the patients' ownership of their healthcare data. This research proposes an architecture for securing data storage and providing efficient access control permission rules between stakeholders viz patients, doctors, pharmacists and other participants via encryption techniques and access control mechanisms. Moreover, a working prototype based on Hyperldger Fabric and Interplanetary File System is made to illustrate the system's viability. The proposed methodology was implemented and evaluated with some use cases for EHRs and consequently, the framework is successful as a reliable health data network. The result of prototype implementation and analysis proves that the approach is a tamper-resistant mechanism as information is stored as hash values for every healthcare transaction in the blockchain. Moreover, it has enormous potential to ensure the privacy, security, integrity, confidentiality and scalability of e-health information. The performance evaluation of the proposed system is completed using empirical research for various scenarios by configuring asset size, block size, various nodes, asset creation time, transaction sets, for evaluation metrics such as transaction latency, transaction throughput, asset latency and data scalability for analysis. The developed POC is shown to be a foolproof system that guarantees the privacy and security of medical information whilst sharing important and sensitive information between stakeholders in a healthcare environment.

This research work also implemented a secure referral system between stakeholders for secure data storage and access to electronic health records utilizing Hyperledger Fabric and Hyperledger Composer (**Chapter 5**). The presented framework and the results of the prototype based on the test cases can be summarized as follows. This research provides a Distributed Ledger Technology Smart contract system for efficient e-Refferal between multiple clinicians in the health data network in the medical industry. This work also created smart contracts for various medical workflows, and then the data access permissions are managed by the patients in the healthcare ecosystem. This research proposes an architecture to secure data storage and provide

efficient access control between stakeholders viz patients, doctors, pharmacists and other participants via encryption techniques and access control mechanisms.

In this research work, a permissioned blockchain framework has been implemented for secure drug prescription tracking between stakeholders in healthcare utilizing Hyperledger Fabric and Hyperledger Composer (**Chapter 6**). This work created smart contracts for medical work flows such as prescription tracking, and the data access permissions are managed by the patient in the healthcare ecosystem. Moreover, this research proposes an efficient cryptographic mechanism to secure data storage and provide efficient access control between stakeholders viz patients, doctors, pharmacists and other participants via encryption techniques and access control mechanisms. A working prototype based on Hyperledger Fabric and Interplanetary File System is made to illustrate the system's viability and consequently, the framework is proven to be successful as a reliable health data network.

This research developed a working prototype based on Hyperledger Fabric and the Interplanetary File System to illustrate the system's viability. The proposed methodology was implemented and evaluated with some use cases for EHRs. Consequently, the framework is proven to be successful as a reliable health data network. The result of prototype implementation and analysis proves that the approach is a tamper-resistant mechanism as information is stored as hash values for every healthcare transaction in the blockchain. Moreover, it has enormous potential to ensure privacy, security, integrity, confidentiality and scalability of the e-health information. Furthermore, this research also explores the technology framework and business processes for blockchain applications. With the volume of healthcare data growing each year, we look forward to improving this prototype with robust scalability simulations and comparing it with other blockchain architectures in a test bed arena that invites more interest in future research work.

The introduction of this technological innovation which incorporates cryptographic elements offers a more secure and effective framework to store, transfer and access EHR in the cloud environment efficiently. The healthchain prototype based on the blockchain technology is a resilient tamperproof ledger as shown by the test results and the POC

rests heavily on the success. With the increase in health data every year, we look forward to refining this prototype with rigorous simulations in scalability and comparing it with other blockchain configurations in a test bed arena that will invite further attention in future research work.

7.2 Study Limitations

Machine Configuration: The healthchain framework is currently a proof of concept that does not completely take into account the complicacy of a true EHR ecosystem. To become a more useful and practical healthcare platform, healthchain can integrate the requirements of healthcare organizations, practitioners, stakeholders and current information systems into the way that it manages and evaluates log data. The proposed system has a few limitations with its system configuration of deploying the framework in a virtual machine environment with limited specifications. Considering the machine configuration, the prototype performed well for the empirical research for various scenarios such as transaction latency, asset latency, transaction throughput, scalability. However, it needs more functionality and functions, as well as more rigorous testing on cloud infrastructures in order to be used by a network of hospitals.

Node Scalability: Another limitation is the node scalability and the proposed framework can be extended to a multiple number of peers and multiple organisations for improved scalability. Due to the incessant increase in health data, it is necessary to effectively increase the peer nodes for efficient storage with the increase in the number of organisations.

Technique Novelty: There are very few proven use cases for the Hyperledger blockchain platform in healthcare. Even though permissioned blockchain platforms are efficient for healthcare, a complete migration to this ecosystem is only possible if all the existing issues are addressed. The are several issues existing with minimum SDKs and less supportive APIs. The ordering node in the proposed framework utilizes kafka which is not completely fault intolerant.

Simulation Environment: The proposed prototype was been implemented and tested in a configured simulated virtual machine environment, therefore when deployed in a real-world environment, the findings obtained from this research may not represent similar results.

Interoperability: There is a demand for open standards to have an interoperable ecosystem between blockchain networks. This research focused more on proof of concept and testing the functionality of blockchain in a configured environment. However, it is important to identify open standards for interoperability requirements for blockchain to be completely implemented and applied in operating healthcare settings.

7.3 Future Research Directions

Blockchain is only in its early years of development in the healthcare sector, which is supported by the fact that the first research literature was published in this sector in 2016. For Healthchain, a significant range of possible research is possible. For instance, both hospital and healthcare providers can create a single blockchain network that can seamlessly move data and offer smooth data exchange between hospitals and stakeholders.

Moreover, we observed from the research study that the scientific contribution to drug prescription management employing blockchain technology is limited in the healthcare sector. To avoid prescription drug abuse, further research can be carried out using various blockchain implementations such as public, private and hybrid to study the advantages and weakness of prescription management systems. In addition, block verification can also be considered as future work. A mechanism to tackle block collision when multiple data blocks arrive at the same time can be studied as future work. The proposed framework is limited by the number of nodes that can be extended to multiple peer nodes and multiple organisations for improved scalability. With the increase in health data every year, we look forward to refining this prototype with rigorous simulations in scalability and comparing it with other blockchain

configurations in a test bed arena that will invite further attention in future research work.

The Digital Asset Modelling Language (DAML) is the language for smart contract running in various ledger platform. As the name implies, it is a modelling language for digital assets, and can work with various databases and ledger technologies can also incorporated as a future work to test its efficiency in the healthcare field.

Our work will improve support for both medical research and government healthcare resource allocation by providing data mining on rich form knowledge and dynamic evolving knowledge, revealing complex, interlinked causal linkages among various factors and providing insights into the trends in and evolutions of the factors.

References

[1] Abbas, A. and Khan, S. U. (2014). A review on the state-of-the-art privacy-preserving approaches in the e-health clouds. *IEEE Journal of Biomedical and Health Informatics*, 18(4):1431–1441.

[2] AbuKhousa, E., Mohamed, N., and Al-Jaroodi, J. (2012). e-health cloud: opportunities and challenges. *Future Internet*, 4(3):621–645.

[3] Adams, C. and Lloyd, S. (1999). *Understanding public-key infrastructure: concepts, standards, and deployment considerations*. Sams Publishing.

[4] Aguilera, M. K. and Toueg, S. (1998). Failure detection and randomization: A hybrid approach to solve consensus. *SIAM Journal on Computing*, 28(3):890–903.

[5] Ahmed, M. and Ullah, A. S. B. (2017). False data injection attacks in healthcare.

[6] Alshehri, S. and Raj, R. K. (2013). Secure access control for health information sharing systems. In *Healthcare Informatics (ICHI), 2013 IEEE International Conference on*, pages 277–286. IEEE.

[7] Anderson, J. C., Lehnardt, J., and Slater, N. (2010). *CouchDB: the definitive guide: time to relax*. " O'Reilly Media, Inc.".

[8] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 30. ACM.

[9] Arriaga, A., Tang, Q., and Ryan, P. (2014). Trapdoor privacy in asymmetric searchable encryption schemes. In *International Conference on Cryptology in Africa*, pages 31–50. Springer.

[10] Ateniese, G., Fu, K., Green, M., and Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30.

[11] Azaria, A., Ekblaw, A., Vieira, T., and Lippman, A. (2016). Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)*, pages 25–30. IEEE.

[12] Baek, J., Safavi-Naini, R., and Susilo, W. (2008). Public key encryption with keyword search revisited. In *International conference on Computational Science and Its Applications*, pages 1249–1259. Springer.

[13] Bahga, A. and Madisetti, V. K. (2013). A cloud-based approach for interoperable electronic health records (ehrs). *IEEE Journal of Biomedical and Health Informatics*, 17(5):894–906.

[14] Baliga, A. (2017). Understanding blockchain consensus models. *Persistent*, 2017(4):1–14.

[15] Barni, M., Failla, P., Lazzeretti, R., Sadeghi, A.-R., and Schneider, T. (2011). Privacy-preserving ecg classification with branching programs and neural networks. *IEEE Transactions on Information Forensics and Security*, 6(2):452–468.

[16] Barua, M., Liang, X., Lu, R., and Shen, X. (2011). Espac: Enabling security and patient-centric access control for ehealth in cloud computing. *International Journal of Security and Networks*, 6(2-3):67–76.

[17] Benet, J. (2014). Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*.

[18] Berghel, H. (2017). Equifax and the latest round of identity theft roulette. *Computer*, 50(12):72–76.

[19] Bessani, A., Sousa, J., and Alchieri, E. E. (2014). State machine replication for the masses with bft-smart. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 355–362. IEEE.

[20] Bethencourt, J., Sahai, A., and Waters, B. (2007). Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 321–334. IEEE.

[21] Bhateja, R., Acharjya, D. P., and Saxena, N. (2017). Enhanced timing enabled proxy re-encryption model for e-health data in the public cloud. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 2040–2044. IEEE.

[22] Biernikiewicz, M., Taieb, V., and Toumi, M. (2019). Characteristics of doctor-shoppers: a systematic literature review. *Journal of market access & health policy*, 7(1):1595953.

[23] Boneh, D., Di Crescenzo, G., Ostrovsky, R., and Persiano, G. (2004). Public key encryption with keyword search. In *International conference on the theory and applications of cryptographic techniques*, pages 506–522. Springer.

[24] Boneh, D., Kushilevitz, E., Ostrovsky, R., and Skeith, W. E. (2007). Public key encryption that allows pir queries. In *Annual International Cryptology Conference*, pages 50–67. Springer.

[25] Brewer, R. (2016). Ransomware attacks: detection, prevention and cure. *Network Security*, 2016(9):5–9.

[26] Calvillo-Arbizu, J., Roman-Martinez, I., and Roa-Romero, L. M. (2014). Standardized access control mechanisms for protecting iso 13606-based electronic health record systems. In *Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on*, pages 539–542. IEEE.

[27] Carpov, S., Nguyen, T. H., Sirdey, R., Constantino, G., and Martinelli, F. (2016). Practical privacy-preserving medical diagnosis using homomorphic encryption. In

Cloud Computing (CLOUD), 2016 IEEE 9th International Conference on, pages 593–599. IEEE.

[28] Cash, D., Jarecki, S., Jutla, C., Krawczyk, H., Roşu, M.-C., and Steiner, M. (2013). Highly-scalable searchable symmetric encryption with support for boolean queries. In *Advances in Cryptology–CRYPTO 2013*, pages 353–373. Springer.

[29] Castro, M., Liskov, B., et al. (1999). Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186.

[30] Charanya, R. and Aramudhan, M. (2016). Survey on access control issues in cloud computing. In *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pages 1–4. IEEE.

[31] Chen, J., Ma, X., Du, M., and Wang, Z. (2018). A blockchain application for medical information sharing. In *2018 IEEE International Symposium on Innovation and Entrepreneurship (TEMS-ISIE)*, pages 1–7. IEEE.

[32] Chen, T.-S., Liu, C.-H., Chen, T.-L., Chen, C.-S., Bau, J.-G., and Lin, T.-C. (2012a). Secure dynamic access control scheme of phr in cloud computing. *Journal of medical systems*, 36(6):4005–4020.

[33] Chen, Y.-Y., Lu, J.-C., and Jan, J.-K. (2012b). A secure ehr system based on hybrid clouds. *Journal of medical systems*, 36(5):3375–3384.

[34] Cheney, J., Chong, S., Foster, N., Seltzer, M., and Vansummeren, S. (2009). Provenance: a future history. In *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, pages 957–964.

[35] Cheng, K., Wang, L., Shen, Y., Wang, H., Wang, Y., Jiang, X., and Zhong, H. (2017). Secure k-nn query on encrypted cloud data with multiple keys. *IEEE Transactions on Big Data*.

[36] Chenthara, S., Ahmed, K., Wang, H., and Whittaker, F. (2019). Security and privacy-preserving challenges of e-health solutions in cloud computing. *IEEE access*, 7:74361–74382.

[37] Chenthara, S., Wang, H., and Ahmed, K. (2018). Security and privacy in big data environment.

[38] Chenthara, S., Wang, H., Ahmed, K., Whittaker, F., and Ji, K. (2020). A blockchain based model for curbing doctors shopping and ensuring provenance management. In *2020 International Conference on Networking and Network Applications (NaNA)*, pages 186–192. IEEE.

[39] Chi, P.-W. and Lei, C.-L. (2018). Audit-free cloud storage via deniable attribute-based encryption. *IEEE Transactions on Cloud Computing*, 6(2):414–427.

[40] Chickowski, E. (2012). Healthcare unable to keep up with insider threats. *Dark Reading (May 2012)*.

[41] Choudhury, A. J., Kumar, P., Sain, M., Lim, H., and Jae-Lee, H. (2011). A strong user authentication framework for cloud computing. In *2011 IEEE Asia-Pacific Services Computing Conference*, pages 110–115. IEEE.

[42] Cui, H., Deng, R. H., and Li, Y. (2018). Attribute-based cloud storage with secure provenance over encrypted data. *Future Generation Computer Systems*, 79:461–472.

[43] Dagher, G. G., Mohler, J., Milojkovic, M., and Marella, P. B. (2018). Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. *Sustainable cities and society*, 39:283–297.

[44] Dannen, C. (2017). *Introducing Ethereum and Solidity*. Springer.

[45] Dhillon, V., Metcalf, D., and Hooper, M. (2017). The hyperledger project. In *Blockchain enabled applications*, pages 139–149. Springer.

[46] Dong, S., Abbas, K., and Jain, R. (2019). A survey on distributed denial of service (ddos) attacks in sdn and cloud computing environments. *IEEE Access*, 7:80813–80828.

[47] Dowell, D., Zhang, K., Noonan, R. K., and Hockenberry, J. M. (2016). Mandatory provider review and pain clinic laws reduce the amounts of opioids prescribed and overdose death rates. *Health Affairs*, 35(10):1876–1883.

[48] Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., and Wang, F. (2017). Secure and trustable electronic medical records sharing using blockchain. In *AMIA Annual Symposium Proceedings*, volume 2017, page 650. American Medical Informatics Association.

[49] Dwivedi, A. D., Srivastava, G., Dhar, S., and Singh, R. (2019). A decentralized privacy-preserving healthcare blockchain for iot. *Sensors*, 19(2):326.

[50] Ekblaw, A., Azaria, A., Halamka, J. D., and Lippman, A. (2016). A case study for blockchain in healthcare:"medrec" prototype for electronic health records and medical research data. In *Proceedings of IEEE open & big data conference*, volume 13, page 13.

[51] El Bouchti, A., Bahsani, S., and Nahhal, T. (2016). Encryption as a service for data healthcare cloud security. In *Future Generation Communication Technologies (FGCT), 2016 Fifth International Conference on*, pages 48–54. IEEE.

[52] Esmaeilzadeh, P. and Mirzaei, T. (2019). The potential of blockchain technology for health information exchange: Experimental study from patients' perspectives. *Journal of medical Internet research*, 21(6):e14184.

[53] Fang, L., Susilo, W., Ge, C., and Wang, J. (2012). Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search. *Theoretical Computer Science*, 462:39–58.

[54] Forrest, C. B., Glade, G. B., Baker, A. E., Bocian, A., von Schrader, S., and Starfield, B. (2000). Coordination of specialty referrals and physician satisfaction with referral care. *Archives of pediatrics & adolescent medicine*, 154(5):499–506.

[55] Fuentes, M. R. (2017). Cybercrime and other threats faced by the healthcare industry. *Trend Micro*.

[56] Gajanayake, R., Iannella, R., and Sahama, T. (2014). Privacy oriented access control for electronic health records. *electronic Journal of Health Informatics*, 8(2):15.

[57] Gentry, C. and Halevi, S. (2011). Implementing gentry's fully-homomorphic encryption scheme. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 129–148. Springer.

[58] Gowda, B. K. and Sumathi, R. (2017). Hierarchy attribute-based encryption with timing enabled privacy preserving keyword search mechanism for e-health clouds. In *Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017 2nd IEEE International Conference on*, pages 425–429. IEEE.

[59] Griebel, L., Prokosch, H.-U., Köpcke, F., Toddenroth, D., Christoph, J., Leb, I., Engel, I., and Sedlmayr, M. (2015). A scoping review of cloud computing in healthcare. *BMC medical informatics and decision making*, 15(1):17.

[60] Guo, L., Zhang, C., Sun, J., and Fang, Y. (2012). Paas: A privacy-preserving attribute-based authentication system for ehealth networks. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, pages 224–233. IEEE.

[61] Hu, V. C., Ferraiolo, D., and Kuhn, D. R. (2006). *Assessment of access control systems*. US Department of Commerce, National Institute of Standards and Technology.

[62] Huang, C., Yan, K., Wei, S., Zhang, G., and Lee, D. H. (2017). Efficient anonymous attribute-based encryption with access policy hidden for cloud computing. In *2017 International Conference on Progress in Informatics and Computing (PIC)*, pages 266–270. IEEE.

[63] Hupperich, T., Löhr, H., Sadeghi, A.-R., and Winandy, M. (2012). Flexible patient-controlled security for electronic health records. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 727–732. ACM.

[64] Ibrahim, A., Mahmood, B., and Singhal, M. (2016). A secure framework for sharing

electronic health records over clouds. In *Serious Games and Applications for Health (SeGAH), 2016 IEEE International Conference on*, pages 1–8. IEEE.

[65] Ivan, D. (2016). Moving toward a blockchain-based method for the secure storage of patient records. In *ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States: ONC/NIST*.

[66] Jamil, F., Ahmad, S., Iqbal, N., and Kim, D.-H. (2020). Towards a remote monitoring of patient vital signs based on iot-based blockchain integrity management platforms in smart hospitals. *Sensors*, 20(8):2195.

[67] Jiang, S., Cao, J., Wu, H., Yang, Y., Ma, M., and He, J. (2018). Blochie: a blockchain-based platform for healthcare information exchange. In *2018 ieee international conference on smart computing (smartcomp)*, pages 49–56. IEEE.

[68] Kaletsch, A. and Sunyaev, A. (2011). Privacy engineering: personal health records in cloud computing environments.

[69] Kamara, S. and Lauter, K. (2010). Cryptographic cloud storage. In *International Conference on Financial Cryptography and Data Security*, pages 136–149. Springer.

[70] Katal, A., Wazid, M., and Goudar, R. H. (2013). Big data: issues, challenges, tools and good practices. In *2013 Sixth international conference on contemporary computing (IC3)*, pages 404–409. IEEE.

[71] Keller, M., Orsini, E., and Scholl, P. (2016). Mascot: faster malicious arithmetic secure computation with oblivious transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 830–842.

[72] Khan, M. F. F. and Sakamura, K. (2015). Fine-grained access control to medical records in digital healthcare enterprises. In *Networks, Computers and Communications (ISNCC), 2015 International Symposium on*, pages 1–6. IEEE.

[73] Khatoon, A. (2020). A blockchain-based smart contract system for healthcare management. *Electronics*, 9(1):94.

[74] King, S. and Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19:1.

[75] Kruse, C. S., Mileski, M., Vijaykumar, A. G., Viswanathan, S. V., Suskandla, U., and Chidambaram, Y. (2017a). Impact of electronic health records on long-term care facilities: Systematic review. *JMIR medical informatics*, 5(3).

[76] Kruse, C. S., Smith, B., Vanderlinden, H., and Nealand, A. (2017b). Security techniques for the electronic health records. *Journal of medical systems*, 41(8):127.

[77] Kuhn, D. R., Coyne, E. J., and Weil, T. R. (2010). Adding attributes to role-based access control. *Computer*, 43(6):79–81.

[78] Lacroix, J. and Boucelma, O. (2014). Trusting the cloud: A prov+ rbac approach. In *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on*, pages 652–658. IEEE.

[79] Lee, W.-B. and Lee, C.-D. (2008). A cryptographic key management solution for hipaa privacy/security regulations. *IEEE Transactions on Information Technology in Biomedicine*, 12(1):34–41.

[80] Leeming, G., Cunningham, J., and Ainsworth, J. (2019). A ledger of me: personalizing healthcare using blockchain technology. *Frontiers in medicine*, 6.

[81] Li, H., Dai, Y., Tian, L., and Yang, H. (2009). Identity-based authentication for cloud computing. In *IEEE international conference on cloud computing*, pages 157–166. Springer.

[82] Li, H., Yang, Y., Dai, Y., Bai, J., Yu, S., and Xiang, Y. (2017a). Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data. *IEEE Transactions on Cloud Computing*.

[83] Li, J., Lin, X., Zhang, Y., and Han, J. (2017b). Ksf-oabe: outsourced attribute-based encryption with keyword search function for cloud storage. *IEEE Transactions on Services Computing*, 10(5):715–725.

[84] Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., and Lou, W. (2010). Fuzzy keyword search over encrypted data in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5. IEEE.

[85] Li, M., Sun, X., Wang, H., Zhang, Y., and Zhang, J. (2011a). Privacy-aware access control with trust management in web service. *World Wide Web*, 14(4):407–430.

[86] Li, M., Yu, S., Zheng, Y., Ren, K., and Lou, W. (2012). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE transactions on parallel and distributed systems*, 24(1):131–143.

[87] Li, M., Yu, S., Zheng, Y., Ren, K., and Lou, W. (2013). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE transactions on parallel and distributed systems*, 24(1):131–143.

[88] Li, P., Guo, S., Miyazaki, T., Xie, M., Hu, J., and Zhuang, W. (2016a). Privacy-preserving access to big data in the cloud. *IEEE Cloud Computing*, 3(5):34–42.

[89] Li, W., Liu, B. M., Liu, D., Liu, R. P., Wang, P., Luo, S., and Ni, W. (2018). Unified fine-grained access control for personal health records in cloud computing. *IEEE journal of biomedical and health informatics*.

[90] Li, W., Xue, K., Xue, Y., and Hong, J. (2016b). Tmacs: A robust and verifiable threshold multi-authority access control system in public cloud storage. *IEEE Transactions on parallel and distributed systems*, 27(5):1484–1496.

[91] Li, Z.-R., Chang, E.-C., Huang, K.-H., and Lai, F. (2011b). A secure electronic medical record sharing mechanism in the cloud computing platform. In *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*, pages 98–103. IEEE.

[92] Liu, P., Wang, J., Ma, H., and Nie, H. (2014). Efficient verifiable public key encryption with keyword search based on kp-abe. In *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2014 Ninth International Conference on*, pages 584–589. IEEE.

[93] Liu, S., Viotti, P., Cachin, C., Quéma, V., and Vukolić, M. (2016). {XFT}: Practical fault tolerance beyond crashes. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 485–500.

[94] Liu, W., Liu, X., Liu, J., Wu, Q., Zhang, J., and Li, Y. (2015). Auditing and revocation enabled role-based access control over outsourced private ehrs. In *High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on*, pages 336–341. IEEE.

[95] Löhr, H., Sadeghi, A.-R., and Winandy, M. (2010). Securing the e-health cloud. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 220–229. ACM.

[96] Ma, M., He, D., Khan, M. K., and Chen, J. (2017). Certificateless searchable public key encryption scheme for mobile healthcare system. *Computers & Electrical Engineering*.

[97] Mahboob, T., Zahid, M., and Ahmad, G. (2016). Adopting information security techniques for cloud computing—a survey. In *2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 7–11. IEEE.

[98] Margheri, A., Masi, M., Miladi, A., Sassone, V., and Rosenzweig, J. (2020). Decentralised provenance for healthcare data. *International Journal of Medical Informatics*, page 104197.

[99] Martins, S. and Yang, Y. (2011). Introduction to bitcoins: a pseudo-anonymous electronic currency system. In *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*, pages 349–350.

[100] Mashima, D. and Ahamad, M. (2012). Enhancing accountability of electronic health record usage via patient-centric monitoring. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 409–418. ACM.

[101] Masud, M. A. H., Huang, X., and Islam, M. R. (2014). A novel approach for the security remedial in a cloud-based e-learning network. *Journal of Networks*, 9(11):2934.

[102] Matturdi, B., Zhou, X., Li, S., and Lin, F. (2014). Big data security and privacy: A review. *China Communications*, 11(14):135–145.

[103] McGraw, D. (2013). Building public trust in uses of health insurance portability and accountability act de-identified data. *Journal of the American Medical Informatics Association*, 20(1):29–34.

[104] Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., and Qijun, C. (2017). A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572. IEEE.

[105] Mohurle, S. and Patil, M. (2017). A brief study of wannacry threat: Ransomware attack 2017. *International Journal of Advanced Research in Computer Science*, 8(5).

[106] Morgan, J. (2016). Quorum whitepaper. *New York: JP Morgan Chase*.

[107] Naehrig, M., Lauter, K., and Vaikuntanathan, V. (2011). Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124. ACM.

[108] Nakamoto, S. (2019). Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot.

[109] Nakamoto, S. et al. (2008). Bitcoin: A peer-to-peer electronic cash system.

[110] Narayan, S., Gagné, M., and Safavi-Naini, R. (2010). Privacy preserving ehr system using attribute-based infrastructure. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, pages 47–52. ACM.

[111] Pecarina, J., Pu, S., and Liu, J.-C. (2012). Sapphire: Anonymity for enhanced control and private collaboration in healthcare clouds. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 99–106. IEEE.

[112] Pramanick, N. and Ali, S. T. (2017). A comparative survey of searchable encryption schemes. In *Computing, Communication and Networking Technologies (ICCCNT), 2017 8th International Conference on*, pages 1–5. IEEE.

[113] Premarathne, U., Abuadbba, A., Alabdulatif, A., Khalil, I., Tari, Z., Zomaya, A., and Buyya, R. (2016). Hybrid cryptographic access control for cloud-based ehr systems. *IEEE Cloud Computing*, 3(4):58–64.

[114] Punithasurya, K. and Jeba Priya, S. (2012). Analysis of different access control mechanism in cloud. *International Journal of Applied Information Systems (IJAIS), Foundation of Computer Science FCS*, 4(2).

[115] Pussewalage, H. S. G. and Oleshchuk, V. (2016a). A patient-centric attribute based access control scheme for secure sharing of personal health records using cloud computing. In *Collaboration and Internet Computing (CIC), 2016 IEEE 2nd International Conference on*, pages 46–53. IEEE.

[116] Pussewalage, H. S. G. and Oleshchuk, V. A. (2016b). An attribute based access control scheme for secure sharing of electronic health records. In *e-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on*, pages 1–6. IEEE.

[117] Pussewalage, H. S. G. and Oleshchuk, V. A. (2017). Attribute based access control scheme with controlled access delegation for collaborative e-health environments. *Journal of Information Security and Applications*, 37:50–64.

[118] Rabieh, K., Akkaya, K., Karabiyik, U., and Qamruddin, J. (2018). A secure and cloud-based medical records access scheme for on-road emergencies. In *Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*, pages 1–8. IEEE.

[119] Roehrs, A., da Costa, C. A., da Rosa Righi, R., da Silva, V. F., Goldim, J. R., and Schmidt, D. C. (2019). Analyzing the performance of a blockchain-based personal health record implementation. *Journal of biomedical informatics*, 92:103140.

[120] Ruj, S., Stojmenovic, M., and Nayak, A. (2012). Privacy preserving access control with authentication for securing data in clouds. In *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*, pages 556–563. IEEE.

[121] Safa, N. S., Sookhak, M., Von Solms, R., Furnell, S., Ghani, N. A., and Herawan, T. (2015). Information security conscious care behaviour formation in organizations. *Computers & Security*, 53:65–78.

[122] Sahai, A. and Waters, B. (2005). Fuzzy identity-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 457–473. Springer.

[123] Sandhu, R., Ferraiolo, D., Kuhn, R., et al. (2000). The nist model for role-based access control: towards a unified standard. In *ACM workshop on Role-based access control*, volume 2000, pages 1–11.

[124] Sandhu, R. S. (1998). Role-based access control. In *Advances in computers*, volume 46, pages 237–286. Elsevier.

[125] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. (1996). Role-based access control models. *Computer*, 29(2):38–47.

[126] Schneberk, T., Raffetto, B., Friedman, J., Wilson, A., Kim, D., and Schriger, D. L. (2020). Opioid prescription patterns among patients who doctor shop; implications for providers. *Plos one*, 15(5):e0232533.

[127] Seol, K., Kim, Y.-G., Lee, E., Seo, Y.-D., and Baik, D.-K. (2018). Privacy-preserving attribute-based access control model for xml-based electronic health record system. *IEEE Access*, 6:9114–9128.

[128] Shao, J., Cao, Z., Liang, X., and Lin, H. (2010). Proxy re-encryption with keyword search. *Information Sciences*, 180(13):2576–2587.

[129] Shen, B., Guo, J., and Yang, Y. (2019). Medchain: Efficient healthcare data sharing via blockchain. *Applied Sciences*, 9(6):1207.

[130] Shi, Y., Liu, J., Han, Z., Zheng, Q., Zhang, R., and Qiu, S. (2014). Attribute-based proxy re-encryption with keyword search. *PloS one*, 9(12):e116325.

[131] Sicuranza, M. and Esposito, A. (2013). An access control model for easy management of patient privacy in ehr systems. In *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for*, pages 463–470. IEEE.

[132] Sukhwani, H., Martínez, J. M., Chang, X., Trivedi, K. S., and Rindos, A. (2017). Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric). In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pages 253–255. IEEE.

[133] Sun, L., Wang, H., Soar, J., and Rong, C. (2012a). Purpose based access control for privacy protection in e-healthcare services. *Journal of Software*, 7(11):2443–2449.

[134] Sun, W., Guo, H., He, H., and Dai, Z. (2007). Design and optimized implementation of the sha-2 (256, 384, 512) hash algorithms. In *2007 7th International Conference on ASIC*, pages 858–861. IEEE.

[135] Sun, X., Li, M., Wang, H., and Plank, A. (2008). An efficient hash-based algorithm for minimal k-anonymity. In *Proceedings of the thirty-first Australasian conference on Computer science-Volume 74*, pages 101–107. Australian Computer Society, Inc.

[136] Sun, X., Wang, H., Li, J., and Zhang, Y. (2012b). Satisfying privacy requirements before data anonymization. *The Computer Journal*, 55(4):422–437.

[137] Sylim, P., Liu, F., Marcelo, A., and Fontelo, P. (2018). Blockchain technology for detecting falsified and substandard drugs in distribution: pharmaceutical supply chain intervention. *JMIR research protocols*, 7.

[138] Tang, Q. and Chen, L. (2009). Public-key encryption with registered keyword search. In *European Public Key Infrastructure Workshop*, pages 163–178. Springer.

[139] Tang, Z., Wei, J., Sallam, A., Li, K., and Li, R. (2012). A new rbac based access control model for cloud computing. In *International Conference on Grid and Pervasive Computing*, pages 279–288. Springer.

[140] Tasatanattakool, P. and Techapanupreeda, C. (2017). User authentication algorithm with role-based access control for electronic health systems to prevent abuse of patient privacy. In *Computer and Communications (ICCC), 2017 3rd IEEE International Conference on*, pages 1019–1024. IEEE.

[141] Thatcher, C. and Acharya, S. (2018). Pharmaceutical uses of blockchain technology. In *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE.

[142] Van Liesdonk, P., Sedghi, S., Doumen, J., Hartel, P., and Jonker, W. (2010). Computationally efficient searchable symmetric encryption. In *Workshop on Secure Data Management*, pages 87–100. Springer.

[143] Vengadapurvaja, A., Nisha, G., Aarthy, R., and Sasikaladevi, N. (2017). An efficient homomorphic medical image encryption algorithm for cloud storage security. *Procedia Computer Science*, 115:643–650.

[144] Venkatram, K. and Geetha, M. A. (2017). Review on big data & analytics–concepts, philosophy, process and applications. *Cybernetics and Information Technologies*, 17(2):3–27.

[145] Vimalananda, V. G., Gupte, G., Seraj, S. M., Orlander, J., Berlowitz, D., Fincke, B. G., and Simon, S. R. (2015). Electronic consultations (e-consults) to improve access to specialty care: a systematic review and narrative synthesis. *Journal of telemedicine and telecare*, 21(6):323–330.

[146] Wang, H. and Song, Y. (2018). Secure cloud-based ehr system using attribute-based cryptosystem and blockchain. *Journal of medical systems*, 42(8):152.

[147] Wang, H., Wang, Y., Taleb, T., and Jiang, X. (2020). Special issue on security and privacy in network computing. *World Wide Web*, 23(2):951–957.

[148] Wang, H., Yi, X., Bertino, E., and Sun, L. (2016). Protecting outsourced data in cloud computing through access management. *Concurrency and computation: Practice and Experience*, 28(3):600–615.

[149] Wang, H., Zhang, Z., and Taleb, T. (2018). Special issue on security and privacy of iot. *World Wide Web*, 21(1):1–6.

[150] Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., and Wang, F.-Y. (2019). Blockchain-enabled smart contracts: architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11):2266–2277.

[151] Wei, J., Liu, W., and Hu, X. (2016). Secure data sharing in cloud computing using revocable-storage identity-based encryption. *IEEE Transactions on Cloud Computing*.

[152] Wood, G. et al. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32.

[153] Xue, K., Xue, Y., Hong, J., Li, W., Yue, H., Wei, D. S., and Hong, P. (2017). Raac: Robust and auditable access control with multiple attribute authorities for public cloud storage. *IEEE Transactions on information Forensics and Security*, 12(4):953–967.

[154] Yang, Y. (2015). Attribute-based data retrieval with semantic keyword search for e-health cloud. *Journal of Cloud Computing*, 4(1):10.

[155] Yang, Y. and Ma, M. (2016). Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds. *IEEE Transactions on Information Forensics and Security*, 11(4):746–759.

[156] Yi, X., Miao, Y., Bertino, E., and Willemson, J. (2013). Multiparty privacy protection for electronic health records. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 2730–2735. IEEE.

[157] Yin, S., Bao, J., Zhang, Y., and Huang, X. (2017). M2m security technology of cps based on blockchains. *Symmetry*, 9(9):193.

[158] Yu, S., Wang, C., Ren, K., and Lou, W. (2010). Achieving secure, scalable, and fine-grained data access control in cloud computing. In *Infocom, 2010 proceedings IEEE*, pages 1–9. Ieee.

[159] Yuan, E. and Tong, J. (2005). Attributed based access control (abac) for web services. In *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*. IEEE.

[160] Yue, X., Wang, H., Jin, D., Li, M., and Jiang, W. (2016). Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, 40(10):218.

[161] Zhang, C., Zhu, L., Xu, C., and Lu, R. (2018a). Ppdp: An efficient and privacy-preserving disease prediction scheme in cloud-based e-healthcare system. *Future Generation Computer Systems*, 79:16–25.

[162] Zhang, M. and Ji, Y. (2018). Blockchain for healthcare records: A data perspective. *PeerJ Preprints*, 6:e26942v1.

[163] Zhang, P., White, J., Schmidt, D. C., and Lenz, G. (2017a). Applying software patterns to address interoperability in blockchain-based healthcare apps. *arXiv preprint arXiv:1706.03700*.

[164] Zhang, P., White, J., Schmidt, D. C., Lenz, G., and Rosenbloom, S. T. (2018b). Fhirchain: applying blockchain to securely and scalably share clinical data. *Computational and structural biotechnology journal*, 16:267–278.

[165] Zhang, R. and Liu, L. (2010). Security models and requirements for healthcare application clouds. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 268–275. IEEE.

[166] Zhang, R., Liu, L., and Xue, R. (2014). Role-based and time-bound access and management of ehr data. *Security and Communication Networks*, 7(6):994–1015.

[167] Zhang, R., Xue, R., and Liu, L. (2017b). Searchable encryption for healthcare clouds: A survey. *IEEE Transactions on Services Computing*.

[168] Zheng, Q., Xu, S., and Ateniese, G. (2014). Vabks: verifiable attribute-based keyword search over outsourced encrypted data. In *Infocom, 2014 proceedings IEEE*, pages 522–530. IEEE.

[169] Zhu, L., Zhang, C., Xu, C., Liu, X., and Huang, C. (2018). An efficient and privacy-preserving biometric identification scheme in cloud computing. *IEEE Access*, 6:19025–19033.

[170] Zyskind, G., Nathan, O., et al. (2015). Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pages 180–184. IEEE.

Appendices

Appendix A

This prototype Implementation for the healthchain network includes several packages, node modules and implementation modules. A few of the main modules are explained below. Appendix.A explains back end files written in Hyperledger Composer Modeling language, Composer query language and Smart contract scripting include Javascript logic for executing the transactions in model file. The front end modules are explained in Appendix.B ie written in Angular 4 programming language. The Angular 4 comprises of component files and main component.ts files are explained here.

A.1 Healthchain.cto

```
//*****************************************************************************
//* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0
//* Author(s) : Shekha Chenthara, Victoria University
//*****************************************************************************
/* Model definitions */

namespace org.ehr.healthchain
```

```
participant Doctor identified by doctorId {
        o String doctorId
        o String firstName
        o String lastName
        o String EmailAddress
        o String pwd
        o String gender
        o String dob default ="yyyy-mm-dd"
        o String pkey
        o String qualifications
        o String specialisation }
participant Pharmacist identified by pharmacistId {
        o String pharmacistId
        o String firstName
        o String lastName
        o String EmailAddress
        o String pwd
        o String gender
        o String dob default ="yyyy-mm-dd"
        o String pkey
        o String qualifications
        o String specialisation }
participant Patient identified by patientId {
        o String patientId
        o String firstName
        o String lastName
        o String EmailAddress
        o String pwd
        o String gender
        o String dob default ="yyyy-mm-dd"
        o String pkey }
```

```
participant Appointment identified by appointmentId {
        o String appointmentId
        --> Patient patientId
        --> Doctor doctorId
        o DateTime AppointmentDateTime
        o YESNO yesno }
participant Chemist identified by chemistId {
        o String chemistId
        o String firstName
        o String lastName
        o String EmailAddress
        o String pwd
        o String gender
        o String dob default ="yyyy-mm-dd"
        o String pkey }
participant Receptionist identified by receptionistId {
        o String receptionistId
        o String firstName
        o String lastName
        o String EmailAddress
        o String pwd
        o String gender
        o String dob default ="yyyy-mm-dd"
        o String pkey }
enum YESNO{
        o ACCEPT
        o REJECT }
enum Permission{
        o READ
        o WRITE
        o DENY }
```

```
enum RoleType{
        o DOCTOR
        o PATIENT
        o CHEMIST
        o RECEPTIONIST
        o PHARMACIST }
asset MedicalRecord identified by recordId{
        o String recordId
        --> Patient patientId
        --> Doctor doctorId
        o String description
        o String recordHash
        o DateTime encounterTime
        o String location }
asset Doctorref identified by recordId{
        o String recordId
        --> Patient patientId
        --> Doctor gpdoctorId
        --> Doctor referdoctorId
        o String description }
asset Prescription identified by recordId{
        o String recordId
        --> Patient patientId
        --> Doctor doctorId
        --> Chemist chemistId
        o String drugdescription
        o Integer quantityPrescribed
        o String recordHash }
transaction UpdateMedicalRecord {
        --> MedicalRecord recordId
        --> Patient patientId
```

```
        --> Doctor GPdoctorId
        --> Doctor referdoctorId
        o String newDescription
        o String newRecordHash
        o DateTime newEncounterTime
    o String newlocation }
asset AddOwnership identified by ownershipId{
        o String ownershipId
        --> MedicalRecord recordId
        o String userId
        o Permission permissionType
        o RoleType roleType }
transaction UpdateOwnership {
        --> AddOwnership ownershipId
        --> MedicalRecord recordId
        o String userId
        o Permission newPermissionType
        o RoleType newRole }
//doctor1 will allow doctor2 to access the medical record
transaction AllowOtherDoctorsRead{
        o String id
        --> MedicalRecord recordId
        --> Doctor doctor2
}
transaction AllowAdoctorWrite{
        --> Patient patientId
        o String doctorId
}
```

A.2 Snippet of Smart Contract File

```
//**************************************************************************
//* Licensed under the Apache License, Version 2.0 (the "License"); you may not use
this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
//* Author(s) : Shekha Chenthara, Victoria University
//**************************************************************************


        'use strict';
        /**  transction processor functions */

 Sample transaction
 @param {org.ehr.healthchain.UpdateOwnership}
 updateOwnership @transaction
*/ async function updateOwnership(tx){
 //const oldValue = tx.asset.value;
 // Update the asset with the new value.
        tx.ownershipId.userId = tx.userId;
        tx.ownershipId.roleType = tx.newRole;
        tx.ownershipId.permissionType = tx.newPermissionType;
// Get the asset registry for the asset.
    const assetRegistry = await getAssetRegistry
    ('org.ehr.healthchain.AddOwnership');
// Update the asset in the asset registry.
        await assetRegistry.update(tx.ownershipId);
// Emit an event for the modified asset.
let event = getFactory().newEvent('org.enexus.ehr', 'SampleEvent');
        event.asset = tx.asset;
        event.oldValue = oldValue;
        event.newValue = tx.newValue;
```

```
        emit(event);
        }
/** Sample transaction
        * @param {org.ehr.healthchain.UpdateMedicalRecord}
        updateMedicalRecord @transaction */
        async function updateMedicalRecord(tx) {
return getAssetRegistry('org.ehr.healthchain.MedicalRecord')
.then(function(assetRegistery) {
 var recordId = updatemedicalrecord.record_Id;
 medicalrecord = getFactory().newResource
 ('org.ehr.healthchain','MedicalRecord', recordId);
         // Update the asset with the new value.
         tx.recordId.location = tx.newlocation;
         tx.recordId.description = tx.newDescription;
         tx.recordId.recordHash = tx.newRecordHash;
         tx.recordId.encounterTime = tx.newEncounterTime;
        // Get the asset registry for the asset.
        const assetRegistry = await getAssetRegistry
        ('org.ehr.healthchain.MedicalRecord');
        // Update the asset in the asset registry.
        await assetRegistry.update(tx.recordId);
        // Emit an event for the modified asset.
        let event = getFactory().newEvent('org.enexus.ehr'
        ,'SampleEvent');
        event.asset = tx.asset;
        event.oldValue = oldValue;
        event.newValue = tx.newValue;
        emit(event); } }
        /**  Sample transaction
        //* @param {org.ehr.healthchain.Doctorref} Doctorref
        * @asset */
```

```
        async function Doctorref(tx) {
          tx.recordId.PatientId = tx.PatientId;
          tx.recordId.GPdoctorID = tx.GPdoctorID;
          tx.recordId.referdoctorID = tx.referdoctorID;
          tx.recordId.description = tx.description;
const assetRegistry = await getAssetRegistry('org.ehr.
healthchain.Doctorref');
          await assetRegistry.update(tx.recordId);
        let event = getFactory().newEvent('', 'SampleEvent');
        event.asset = tx.asset;
        event.oldValue = oldValue;
        event.newValue = tx.newValue;
        emit(event);}
/*** Allow a chemist to access a record
        * @param {org.ehr.healthchain.AllowChemistRead} allowAccess
        * @return {Promise} Asset Registry Promise @transaction */
async function AllowChemistRead(allowReadAccess){
return getAssetRegistry('org.ehr.healthchain.Prescription')
.then(function(assetRegistery) {
   record.AllowChemistRead = Prescription.chemistId;
let event = getFactory().newEvent('org.enexus.ehr',
'Prescription_recordId');
   event.asset = tx.asset;
   event.oldValue = oldValue;
   event.newValue = tx.newValue;
   emit(event);
return assetRegistery.update
(allowChemistAccess.Prescription_recordId);      }

/*** Allow a doctor to access a record @param
{org.ehr.healthchain.AllowOtherDoctorsRead}
```

```
allowDoctorAccess @return {Promise}
Asset Registry Promise @transaction*/
        async function allowDoctor(allowDoctorAccess){
        var id=allowDoctorAccess.id;
        var doctor2_id=allowDoctorAccess.doc2.DoctorId;
        return getAssetRegistry('org.ehr.healthchain.Medical_Record').
        then(function(assetRegistery) {
                if(id==allowDoctorAccess.record.DoctorId){
                  allowDoctorAccess.record.version++;
                  allowDoctorAccess.record.authorized
                 .push(doctor2_id);
                  return assetRegistery
                 .update(allowDoctorAccess.record);}
          else if(id==allowDoctorAccess.record.PatientId){
                allowDoctorAccess.record.version++;
                allowDoctorAccess.record.authorized.push(doctor2_id);
                return assetRegistery.update(allowDoctorAccess.record);
        else{
 for(var i=0;i<allowDoctorAccess.record.authorized.length;i++){
        if(allowDoctorAccess.record.authorized[i]==id){
        allowDoctorAccess.record.version++;
        allowDoctorAccess.record.authorized.push(doctor2_id);
        return assetRegistery.update(allowDoctorAccess.record);
  } } } throw "Too big";         });      }
```

A.3 Query File

```
/** A snippet of the query file **/
query selectPatients {
        description: "Select all Patients"
```

```
        statement:
        SELECT org.ehr.healthchain.Patient}
query selectPatientById {
        description: "Select the patient based on their id"
        statement:
        SELECT org.ehr.healthchain.Patient
        WHERE (patientId ==_$PatientId)}
query selectPatientByEmailandpwd {
        description: "Select the patient based on their email and pwd"
        statement:
        SELECT org.ehr.healthchain.Patient
        WHERE ((EmailAddress ==_$EmailAddress) AND (pwd==_$pwd))}
query selectDoctors {
        description: "Select all doctors"
        statement:
        SELECT org.ehr.healthchain.Doctor}
query selectDoctorById {
        description: "Select the doctor based on their id"
        statement:
        SELECT org.ehr.healthchain.Doctor
        WHERE (doctorId ==_$DoctorId)}
query selectDoctorByEmailandpwd {
        description: "Select the doctor based on their email and pwd"
        statement:
        SELECT org.ehr.healthchain.Doctor
        WHERE ((EmailAddress ==_$EmailAddress) AND (pwd==_$pwd))}
query selectChemistById {
        description:"Select the chemist based on their id"
        statement:
        SELECT org.ehr.healthchain.Chemist
        WHERE (chemistId ==_$chemistId)}
```

```
query selectChemistByEmailandpwd {
        description:"Select the chemist based on their email and pwd"
        statement:
        SELECT org.ehr.healthchain.Chemist
        WHERE ((EmailAddress ==_$EmailAddress) AND (pwd==_$pwd))}
query selectReceptionistByEmailandpwd {
        description:"Select the receptionist based on their email
         and pwd"
        statement:
        SELECT org.ehr.healthchain.Receptionist
        WHERE ((EmailAddress ==_$EmailAddress) AND (pwd==_$pwd))}
query selectMedicalRecordByDoctorId {
        description: "Select the medical records based on
        the DoctorId"
        statement:
        SELECT org.ehr.healthchain.MedicalRecord
        WHERE (doctorId ==_$DoctorId )}
query selectMedicalRecordByIPFSHASH {
        description: "Select the medical records based
        on the ipfsHash"
        statement:
        SELECT org.ehr.healthchain.MedicalRecord
        WHERE (recordHash ==_$recordHash)}
query selectMedicalRecordByPatientId {
        description: "Select the medical records
        based on the PatientId"
        statement:
        SELECT org.ehr.healthchain.MedicalRecord
        WHERE (patientId ==_$PatientId)}
query selectMedicalRecordByDoctorAndPatientId {
        description: "Select the medical records
```

```
        based on the DoctorId and PatientId"
        statement:
        SELECT org.ehr.healthchain.MedicalRecord
        WHERE ((doctorId ==_$DoctorId) AND
        (patientId == _$PatientId))}
query selectMedicalRecordByDoctorAndPatientIdAndTime {
        description: "Select the medical records based on the
        DoctorId and PatientId sorted by time"
        statement:
        SELECT org.ehr.healthchain.MedicalRecord
        WHERE ((doctorId ==_$DoctorId) AND (patientId == _$PatientId))
        ORDER BY encounterTime}
query selectDoctorrefByGPDoctorID {
        description: "Select the referral records
        based on the GPDoctorId "
        statement:
        SELECT org.ehr.healthchain.Doctorref
        WHERE ((gpdoctorId ==_$DoctorId)) }
query selectDoctorrefByreferDoctorID {
        description: "Select the referral records
        based on the referDoctorId "
        statement:
        SELECT org.ehr.healthchain.Doctorref
        WHERE ( (referdoctorId ==_$DoctorId))}
query selectPrescriptionBychemistID {
        description: "Select the chemist records
         based on recordID"
        statement:
        SELECT org.ehr.healthchain.Prescription
        WHERE ((chemistId == _$ChemistID))
}
```

```
query selectPrescriptionByDoctorID {
        description: "Select the pharmacist records
        based on recordID"
        statement:
        SELECT org.ehr.healthchain.Prescription
        WHERE ((doctorId == _$DoctorID))
}
query selectOwnershipById {
        description: "Select the ownership by Id"
        statement:
        SELECT org.ehr.healthchain.AddOwnership
        WHERE (ownershipId ==_$ownershipId)}
query selectOwnershipByrecordId {
        description: "Select the record by Id"
        statement:
        SELECT org.ehr.healthchain.AddOwnership
        WHERE (recordId ==_$recordId)}
query selectOwnershipByuserIdandRole {
        description: "Select the ownership by user Id and role"
        statement:
        SELECT org.ehr.healthchain.AddOwnership
        WHERE ((userId ==_$userId) AND (roleType==_$roleType))}
```

Appendix B

B.1 SignUpForm.Component.ts

```
//*********************************************************************
//* Licensed under the Apache License, Version 2.0 (the "License"); you may not use
this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
//* Author(s) : Shekha Chenthara, Victoria University
//*********************************************************************

import { Component, OnInit } from '@angular/core';
import { Router, ActivatedRoute } from '@angular/router';
import { FormBuilder, FormGroup,FormControl, Validators }
from '@angular/forms';
import { DoctorService } from '../Doctor/Doctor.service';
import { PatientService } from '../Patient/Patient.service';
import { ChemistService } from '../Chemist/Chemist.service';
import { ReceptionistService } from '
../Receptionist/Receptionist.service';
```

```
@Component({
selector: 'app-signup',templateUrl: './SignUpForm.component.html'
        ,providers: [DoctorService,PatientService,ChemistService,
        ReceptionistService]
})
export class SignUpFormComponent implements OnInit {
        private participant;
        SignUpForm: FormGroup;
        loading = false;
        submitted = false;
        returnUrl: string;
        error = '';
        doctorId = new FormControl('', Validators.required);
        firstName = new FormControl('', Validators.required);
        lastName = new FormControl('', Validators.required);
        EmailAddress = new FormControl('', Validators.required);
        password = new FormControl('', Validators.required);
        age = new FormControl('', Validators.required);
        gender = new FormControl('', Validators.required);
        userTypes: string[] = ['Doctor', 'Patient', 'Chemist',
        'Receptionist'];
        default: string = 'Patient';
        user=new FormControl('',Validators.required);

        constructor(
        private formBuilder: FormBuilder,
        private route: ActivatedRoute,
        private router: Router
        ,private Doctorcomp: DoctorService
        ,private Patientcomp: PatientService
        ,private Chemistcomp: ChemistService
```

```
,private Receptionistcomp: ReceptionistService
//private authenticationService: AuthenticationService
) {
        this.SignUpForm = formBuilder.group({
                doctorId: this.doctorId,
                firstName: this.firstName,
                lastName: this.lastName,
                EmailAddress: this.EmailAddress,
                password: this.password,
                age: this.age,
                gender: this.gender,
                user:this.user
        });}
ngOnInit() {
        console.log("inside 11);");}
onSubmit() {
this.submitted = true;
console.log(this.user.value);
if(this.user.value=='Doctor'){
        this.participant = {
                $class: 'org.ehr.healthchain.Doctor',
                'doctorId': this.doctorId.value,
                'firstName': this.firstName.value,
                'lastName': this.lastName.value,
                'EmailAddress': this.EmailAddress.value,
                'pwd': this.password.value,
                'gender': this.gender.value,
                'dob': this.age.value,
                'pkey': "No key",
                'qualifications': "Enter Qualifications",
                'specialisation': "Enter Specialisation"};
```

```
            return this.Doctorcomp.addParticipant(this.participant)
            .toPromise().then(() => {
                        this.router.navigate(['/']);     })
    if(this.user.value=='Patient'){
                console.log("inside patient");
                this.participant = {
                        $class: 'org.ehr.healthchain.Patient',
                        'patientId': this.doctorId.value,
                        'firstName': this.firstName.value,
                        'lastName': this.lastName.value,
                        'EmailAddress': this.EmailAddress.value,
                        'pwd': this.password.value,
                        'gender': this.gender.value,
                        'dob': this.age.value,
                        'pkey': "no key" };
        return this.Patientcomp.addParticipant(this.participant)
                        .toPromise()
                        .then(() => {
                        this.router.navigate(['/']);     });      }
        if(this.user.value=='Chemist'){
                console.log("inside chemist");
                this.participant = {
                        $class: 'org.ehr.healthchain.Chemist',
                        'chemistId': this.doctorId.value,
                        'firstName': this.firstName.value,
                        'lastName': this.lastName.value,
                        'EmailAddress': this.EmailAddress.value,
                        'pwd': this.password.value,
                        'gender': this.gender.value,
                        'dob': this.age.value,
                        'pkey': "no key"                 };
```

```
        return this.Chemistcomp.addParticipant(this.participant)
                        .toPromise()
                        .then(() => {
                                this.router.navigate(['/']);
                        });     }
        if(this.user.value=='Receptionist'){
                console.log("inside receptionist");
                this.participant = {
                        $class: 'org.ehr.healthchain.Receptionist',
                        'receptionistId': this.doctorId.value,
                        'firstName': this.firstName.value,
                        'lastName': this.lastName.value,
                        'EmailAddress': this.EmailAddress.value,
                        'pwd': this.password.value,
                        'gender': this.gender.value,
                        'dob': this.age.value,
                        'pkey': "No key"
                        };
        return this.Receptionistcomp.addParticipant(this.participant)
                        .toPromise()
                        .then(() => {
                        this.router.navigate(['/']);}); }}}
```

B.2 MedicalRecord.Component.ts

```
//*****************************************************************
```

//* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

//* Author(s) : Shekha Chenthara, Victoria University

```
//*********************************************************************

import { Component, OnInit, Input } from '@angular/core';
import { FormGroup, FormControl, Validators, FormBuilder }
from '@angular/forms';
import { MedicalRecordService } from './MedicalRecord.service';
import 'rxjs/add/operator/toPromise';
import $ from 'jquery';
import {Buffer} from 'buffer';
import IpfsApi from 'ipfs-api';
import * as CryptoJS from 'crypto-js';
import {ActivatedRoute} from '@angular/router';
import {DataService} from '../data.service';
import { HttpClient, HttpEvent, HttpEventType }
from '@angular/common/http';
import { buffer } from 'rxjs/operator/buffer';
import { BoundCallbackObservable }
from 'rxjs/observable/BoundCallbackObservable';
import { async } from 'q';
import { elementAt } from 'rxjs/operator/elementAt';
//import IPFSUploader from 'ipfs-image-web-upload';
@Component({
selector: 'app-medicalrecord',
templateUrl: './MedicalRecord.component.html',
styleUrls: ['./MedicalRecord.component.css'],
providers: [MedicalRecordService]
})
export class MedicalRecordComponent implements OnInit {
myForm: FormGroup;
private allAssets;
private asset;
```

```
private currentId;
private errorMessage;
ipfs:IpfsApi;
public hash: string;
name = '';
role='';
id: string;
data1:any;
Url: string;
length:number;
objectLength:number;
content:string='';
write:string='';
data3:any;

recordId = new FormControl('', Validators.required);
patientId = new FormControl('', Validators.required);
doctorId = new FormControl('', Validators.required);
description = new FormControl('', Validators.required);
recordHash = new FormControl('', Validators.required);
encounterTime = new FormControl('', Validators.required);
location = new FormControl('', Validators.required);
//imageURL = new FormControl('', Validators.required);
constructor( private route: ActivatedRoute,
private user: DataService<any>,
private _http: HttpClient,
public serviceMedicalRecord: MedicalRecordService, fb: FormBuilder){
this.myForm = fb.group({
recordId: this.recordId,
patientId: this.patientId,
doctorId: this.doctorId,
```

```
description: this.description,
recordHash: this.recordHash,
encounterTime: this.encounterTime,
location: this.location
//imageURL : this.imageURL
});
this.bootstapIpfs();
this.saveToIpfs = this.saveToIpfs.bind(this);
};

ngOnInit(): void {
console.log('Is user logged in? ', this.user.getUserLoggedIn()+"
used id:"+ this.user.getId())
console.log("inside Patient");
console.log("name"+this.user.getUsername());
this.name = this.user.getUsername();
this.role=this.user.getRole();
this.id=this.user.getId();
this.loadAll();}
bootstapIpfs()
{
this.ipfs = new IpfsApi({host: 'ipfs.infura.io', port: 5001,
protocol: 'https'});}
public async saveToIpfs (files)
{
console.log(files)
event.stopPropagation()
event.preventDefault()
this.uploadtoipfs(files,async (arrayBuffer)=>
{
console.log('returned',arrayBuffer)
```

```
const content = Buffer.from(arrayBuffer)
console.log('content',content);
const filesAdded = await this.ipfs.files.add(content)
console.log("filesadded",filesAdded)
this.recordHash=filesAdded[0].hash;
console.log(this.recordHash)
let url = 'https://ipfs.io/ipfs/'.concat(filesAdded[0].hash);
console.log('url',url);
document.getElementById('output').setAttribute('src',url);
});}
public uploadtoipfs(files,callback){
const reader = new FileReader()
reader.readAsArrayBuffer(files[0])
console.log("Buffering...");
var arrayBuffer=reader.onload = function ()
{
var arrayBuffer = reader.result;
console.log("Buffer: ")
console.log(arrayBuffer);
callback(arrayBuffer);
}}
setEncrypt(keys, value){
---
return encrypted.toString();
}
getDecrypt(keys, value){
---
return decrypted.toString(CryptoJS.enc.Utf8);
}
public async set(path: string, value: string) {
}
```

```
public async get(hash: string) {
console.log(hash)
const fileBuffer = await this.ipfs.files.cat(hash);
var decrypted = this.getDecrypt('', fileBuffer.toString());
console.log('Decrypted :' + decrypted);
console.log(fileBuffer.toString());
}
if(this.role=='Doctor'){
this.Url='http://localhost:3000/api/queries/selectMedicalRecordBy
DoctorId?DoctorId=resource%3Aorg.ehr.healthchain.Doctor%23'+this.id
}
if(this.role=='Doctor')
{
this.data1.forEach(asset => {
console.log("recordids:"+asset.recordId)
this._http.get('http://localhost:3000/api/queries/
selectOwnershipByrecordId?recordId'+
'=resource%3Aorg.ehr.healthchain.MedicalRecord%23'+asset.recordId)
.subscribe((data2: any) => {
this.data3=data2;
if(data2[0] === undefined){
console.log("undefined:"+asset.recordId)
tempList1.push(asset);}
if((data2.length>0) &&(data2[0].permissionType=='READ'||
data2[0].permissionType=='WRITE')||data2[0] === undefined)
{console.log("inside");
if(data2[0].permissionType=='WRITE'){
this.write='yes';
console.log("yes");
}tempList1.push(asset);
}});});
```

```
addAsset(form: any): Promise<any> {
console.time("test")
this.asset = {
$class: 'org.ehr.healthchain.MedicalRecord',
'recordId': this.recordId.value,
'patientId': this.patientId.value,
'doctorId': this.doctorId.value,
'description': this.description.value,
'recordHash': this.recordHash,
'encounterTime': this.encounterTime.value,
'location': this.location.value
};
---
return this.serviceMedicalRecord.addAsset(this.asset)
.toPromise()
.then(() => {
this.errorMessage = null;
this.myForm.setValue({
'recordId': null,
'patientId': null,
'doctorId': null,
'description': null,
'recordHash': null,
'encounterTime': null,
'location': null
});
this.loadAll();
})
updateAsset(form: any): Promise<any> {
this.asset = {
$class: 'org.ehr.healthchain.MedicalRecord',
```

```
---
this.loadAll();
})
.catch((error) => {
if (error === 'Server error') {
this.errorMessage = 'Could not connect to REST server.
Please check your configuration details';
} else if (error === '404 - Not Found') {
this.errorMessage = '404 - Could not find API route.
Please check your available APIs.';
} else {
this.errorMessage = error;
}});}
```

B.3 Doctorref.Component.ts

```
import { Component, OnInit, Input } from '@angular/core';
import { FormGroup, FormControl, Validators, FormBuilder }
from '@angular/forms';
import { MedicalRecordService } from './Doctorref.service';
import 'rxjs/add/operator/toPromise';
import $ from 'jquery';
import {Buffer} from 'buffer';
import IpfsApi from 'ipfs-api';
import * as CryptoJS from 'crypto-js';
import {ActivatedRoute} from '@angular/router';
import {DataService} from '../data.service';
import { HttpClient, HttpEvent, HttpEventType }
from '@angular/common/http';
import { buffer } from 'rxjs/operator/buffer';
```

```
import { async } from 'q';
import { elementAt } from 'rxjs/operator/elementAt';
@Component({
selector: 'app-doctorref',
templateUrl: './Doctorref.component.html',
styleUrls: ['./Doctorref.component.css'],
providers: [MedicalRecordService]
})
export class DoctorrefComponent implements OnInit {
myForm: FormGroup;
private allAssets;
private asset;
private currentId;
private errorMessage;
ipfs:IpfsApi;
public hash: string;
name = '';
role='';
id: string;
data1:any;
Url: string;
length:number;
objectLength:number;
content:string='';
write:string='';
data3:any;
recordId = new FormControl('', Validators.required);
patientId = new FormControl('', Validators.required);
gpdoctorId = new FormControl('', Validators.required);
referdoctorId = new FormControl('', Validators.required);
description = new FormControl('', Validators.required);
```

```
constructor( private route: ActivatedRoute,private user:
DataService<any>,
private _http: HttpClient,
public serviceMedicalRecord: MedicalRecordService,
public fb: FormBuilder) {
this.myForm = fb.group({
recordId: this.recordId,
patientId: this.patientId,
gpdoctorId: this.gpdoctorId,
referdoctorId: this.referdoctorId,
description: this.description, });
 };
 loadAll() {
 if(this.role=='Patient'){
 this.Url='http://localhost:3000/api/queries/selectMedicalRecordBy
 PatientId?PatientId=resource%3Aorg.ehr.healthchain.Patient%23'
 +this.id }
 if(this.role=='Doctor')
 {
 this.Url='http://localhost:3000/api/queries/selectDoctorrefBy
 ReferDoctorID?DoctorId=resource%3Aorg.ehr.healthchain.Doctor%23'
 +this.id }
---
 this.allAssets = tempList1;
 }
 if(this.role=='Doctor')
 {
 this.data1.forEach(asset => {
 console.log('asset value: ',asset)
 console.log("recordids:"+asset.recordId)
 this._http.get('http://localhost:3000/api/queries/
```

```
 selectOwnershipByrecordId?recordId'+
 '=resource%3Aorg.ehr.healthchain.MedicalRecord%23'+asset.recordId)
 .subscribe((data2: any) => {
 this.data3=data2;
  if(data2[0] === undefined)
 {
 console.log("undefined:"+asset.recordId)
 //console.log("perm type::"+data2[0].permissionType)
 tempList1.push(asset);
 }
---
if(this.role=='Doctor') {
 this.Url='http://localhost:3000/api/queries/selectDoctorrefBy
 GPDoctorID?DoctorId=resource%3Aorg.ehr.healthchain.Doctor%23'
 +this.id }
 this._http.get(this.Url)
 .subscribe((data: any) => {
 this.data1=data;
 console.log(data);
 this.length=this.data1.length;
 console.log('len:',this.length);
 const tempList1 = [];
 if(this.length >0){
 console.log("array");

 if(this.role=='Patient'){
 this.data1.forEach(asset => {
 tempList1.push(asset);
 console.log("templist1:"+tempList1)
 });
 this.allAssets = tempList1;
```

```
}
if(this.role=='Doctor')
{
this.data1.forEach(asset => {
console.log('asset value: ',asset)
console.log("recordids:"+asset.recordId)
this._http.get('http://localhost:3000/api/queries/
selectOwnershipByrecordId?recordId'+ '=resource%3Aorg.ehr
.healthchain.MedicalRecord%23'+asset.recordId)
.subscribe((data2: any) => {
this.data3=data2;
if(data2[0] === undefined) {
console.log("undefined:"+asset.recordId)
tempList1.push(asset); }
if((data2.length>0) &&(data2[0].permissionType=='READ'||
data2[0].permissionType=='WRITE')
||data2[0] === undefined)
{console.log("inside");
if(data2[0].permissionType=='WRITE'){
this.write='yes';
console.log("yes");
}
tempList1.push(asset);
});
this.allAssets = tempList1;
}
---
addAsset(form: any): Promise<any> {
console.time("test")
this.asset = {
$class: 'org.ehr.healthchain.Doctorref',
```

```
'recordId': this.recordId.value,
'patientId': this.patientId.value,
'gpdoctorId': this.gpdoctorId.value,
'referdoctorId': this.referdoctorId.value,
'description': this.description.value
};
 return this.serviceMedicalRecord.addAsset(this.asset)
.toPromise()
.then(() => {
this.errorMessage = null;
this.myForm.setValue({
'recordId': null,
'patientId': null,
'gpdoctorId': null,
'referdoctorId': null,
'description': null
 });
 this.loadAll();
})
.catch((error) => {
if (error === 'Server error') {
this.errorMessage = 'Could not connect to REST server.
Please check your configuration details';
} else {
this.errorMessage = error;
}); }
referralForm(): void {
this.myForm.setValue({
'patientId': null,
'doctorId': null,
'description': null,
```

```
/*'recordHash': null,
'encounterTime': null,
'location': null*/
}); }
```

B.4 Prescription.Component.ts

```
@Component({
selector: 'app-Prescription',
templateUrl: './Prescription.component.html',
styleUrls: ['./Prescription.component.css'],
providers: [PrescriptionService]
})
export class PrescriptionComponent implements OnInit {
myForm: FormGroup;
---
recordId = new FormControl('', Validators.required);
patientId = new FormControl('', Validators.required);
doctorId = new FormControl('', Validators.required);
chemistId = new FormControl('', Validators.required);
drugdescription = new FormControl('', Validators.required);
quantityPrescribed = new FormControl('', Validators.required);
recordHash = new FormControl('', Validators.required);
constructor(private route: ActivatedRoute,
private user: DataService<any>,
private _http: HttpClient,
public servicePrescription: PrescriptionService,
fb: FormBuilder) {
this.myForm = fb.group({
recordId: this.recordId,
```

```
patientId: this.patientId,
doctorId: this.doctorId,
chemistId: this.chemistId,
drugdescription: this.drugdescription,
quantityPrescribed: this.quantityPrescribed,
recordHash: this.recordHash
});
this.bootstapIpfs();
this.saveToIpfs = this.saveToIpfs.bind(this);
};
ngOnInit(): void {
console.log('Is user logged in? ', this.user.getUserLoggedIn()+
" used id:"+ this.user.getId())
console.log("inside Patient");
console.log("name"+this.user.getUsername());
this.name = this.user.getUsername();
this.role=this.user.getRole();
this.id=this.user.getId();
this.loadAll();}
 bootstapIpfs()
{
this.ipfs = new IpfsApi({host: 'ipfs.infura.io', port: 5001,
protocol: 'https'});}
public async saveToIpfs (files)
{
console.log(files)
event.stopPropagation()
event.preventDefault()
this.uploadtoipfs(files,async (arrayBuffer)=>
{
console.log('returned',arrayBuffer)
```

```
const content = Buffer.from(arrayBuffer)
console.log('content',content);
const filesAdded = await this.ipfs.files.add(content)
console.log("filesadded",filesAdded)
this.recordHash=filesAdded[0].hash;
console.log(this.recordHash)
let url = 'https://ipfs.io/ipfs/'.concat(filesAdded[0].hash);
console.log('url',url);
document.getElementById('output').setAttribute('src',url);
});}
public uploadtoipfs(files,callback){
const reader = new FileReader()
reader.readAsArrayBuffer(files[0])
console.log("Buffering...");
var arrayBuffer=reader.onload = function ()
{
var arrayBuffer = reader.result;
console.log("Buffer: ")
console.log(arrayBuffer);
callback(arrayBuffer);
}}
loadAll() {
if(this.role=='Chemist'){
this.Url='http://localhost:3000/api/queries/selectPrescriptionBy
chemistID?ChemistID=resource%3Aorg.ehr.healthchain.Chemist%23'+this.id
}
if(this.role=='Doctor'){
this.Url='http://localhost:3000/api/queries/selectPrescriptionBy
DoctorID?DoctorID=resource%3Aorg.ehr.healthchain.Doctor%23'+this.id
}
this._http.get(this.Url)
```

```
.subscribe((data: any) => {
this.data1=data;
console.log(data);
const tempList1 = [];
if(this.data1 instanceof Array){
console.log("array");
this.length=this.data1.length;
if(this.role=='Chemist'){
this.data1.forEach(asset => {
tempList1.push(asset);
console.log("templist1:"+tempList1)
});
this.allAssets = tempList1;
}
if(this.role=='Doctor')
{
this.data1.forEach(asset => {
console.log("recordids:"+asset.recordId)
this._http.get('http://localhost:3000/api/queries/
selectOwnershipByrecordId?recordId'+'=resource%3Aorg.
ehr.healthchain.Prescription%23'+ asset.recordId)
.subscribe((data2: any) => {
this.data3=data2;
if(data2[0] === undefined)
{
console.log("undefined:"+asset.recordId)
tempList1.push(asset);
}
 if((data2.length>0) &&(data2[0].permissionType=='READ'||
 data2[0].permissionType=='WRITE')||data2[0] === undefined)
{console.log("inside");
```

```
if(data2[0].permissionType=='WRITE'){
this.write='yes';
console.log("yes");
}
tempList1.push(asset);
}});});
this.allAssets = tempList1;
}}});
changeArrayValue(name: string, value: any): void {
const index = this[name].value.indexOf(value);
if (index === -1) {
this[name].value.push(value);
} else {
this[name].value.splice(index, 1);
}}
addAsset(form: any): Promise<any> {
console.time("test")
this.asset = {
$class: 'org.ehr.healthchain.Prescription',
'recordId': this.recordId.value,
'patientId': this.patientId.value,
'doctorId': this.doctorId.value,
'chemistId': this.chemistId.value,
'drugdescription': this.drugdescription.value,
'quantityPrescribed': this.quantityPrescribed.value,
'recordHash': this.recordHash
};

this.myForm.setValue({
'recordId': null,
'patientId': null,
```

```
'doctorId': null,
'chemistId': null,
'drugdescription': null,
'quantityPrescribed': null,
'recordHash': null
});

return this.servicePrescription.addAsset(this.asset)
.toPromise()
.then(() => {
this.errorMessage = null;
this.myForm.setValue({
'recordId': null,
'patientId': null,
'doctorId': null,
'chemistId': null,
'drugdescription': null,
'quantityPrescribed': null,
'recordHash': null
});
this.loadAll();
})
.catch((error) => {
if (error === 'Server error') {
this.errorMessage = 'Could not connect to REST server.
 Please check your configuration details';
} else {
this.errorMessage = error;
}});}
updateAsset(form: any): Promise<any> {
this.asset = {
```

```
$class: 'org.ehr.healthchain.Prescription',
'recordId': this.recordId.value,
'patientId': this.patientId.value,
'doctorId': this.doctorId.value,
'chemistId': this.chemistId.value,
'drugdescription': this.drugdescription.value,
'quantityPrescribed': this.quantityPrescribed.value,
'recordHash': this.recordHash
};
return this.servicePrescription.updateAsset
(form.get('recordId').value, this.asset)
.toPromise()
.then(() => {
this.errorMessage = null;
this.loadAll();
})
.catch((error) => {
if (error === 'Server error') {
this.errorMessage = 'Could not connect to REST server.
Please check your configuration details';
} else if (error === '404 - Not Found') {
this.errorMessage = '404 - Could not find API route.
Please check your available APIs.';
} else {
this.errorMessage = error;
}});}
deleteAsset(): Promise<any> {
return this.servicePrescription.deleteAsset(this.currentId)
.toPromise()
.then(() => {
this.errorMessage = null;
```

```
this.loadAll();
})
.catch((error) => {
if (error === 'Server error') {
this.errorMessage = 'Could not connect to REST server.
Please check your configuration details';
} else if (error === '404 - Not Found') {
this.errorMessage = '404 - Could not find API route.
Please check your available APIs.';
} else {
this.errorMessage = error;
}});}
setId(id: any): void {
this.currentId = id;
}
getForm(id: any): Promise<any> {
return this.servicePrescription.getAsset(id)
.toPromise()
.then((result) => {
this.errorMessage = null;
const formObject = {
'recordId': null,
'patientId': null,
'doctorId': null,
'chemistId': null,
'drugdescription': null,
'quantityPrescribed': null,
'recordHash': null
};
---
this.myForm.setValue(formObject);
```

```
})
.catch((error) => {
if (error === 'Server error') {
this.errorMessage = 'Could not connect to REST server.
Please check your configuration details';
} else if (error === '404 - Not Found') {
this.errorMessage = '404 - Could not find API route.
Please check your available APIs.';
} else {
this.errorMessage = error;
}});}
resetForm(): void {
this.myForm.setValue({
'recordId': null,
'patientId': null,
'doctorId': null,
'chemistId': null,
'drugdescription': null,
'quantityPrescribed': null,
'recordHash': null
});}}
```