Università degli Studi di Camerino

School of Advanced Studies

Doctoral course in

BLOCKCHAIN AND DISTRIBUTED LEDGER TECHNOLOGY

Curriculum

Healthcare and Well-being

Cycle

XXXVIII

Protecting Healthcare Data: Blockchain-based Solutions for Security, Privacy and Scalability

Ph.D. Student:

Jahan Zeb Shahid

Supervisor:

Prof. Stelvio Cimato

Co-supervisors:

Prof. Muslim Jameel Syed

Coordinator of the Ph.D. Program:

Prof. Flavio Corradini *(UniCam)*

Date of award: **23/06/2026**

Abstract

Every individual in our digitally advanced world requires adequate data protection. The transmission of healthcare information is increasingly focused on protecting data privacy and improving scalability. The rapid progression of communications technology has made the sharing of distributed information across several domains an inescapable trend, enhancing transmission of information with an emphasis on privacy and scalability.

The fast expansion of blockchain systems sometimes requires the substitution of customized blockchains with more secure, scalable, and decentralized alternatives. Ensuring privacy in healthcare data is particularly tough due to stringent data laws and regulations. Permissioned blockchain solutions are gaining prominence due to their transparency, immutability, and customization features. A consortium-based permissioned blockchain provides better security to healthcare data. However, protecting privacy is crucial and demanding.

Federated learning (FL) ensures data privacy as the federated devices retain the real data locally without sharing it. It serves as an advantageous instrument when used with blockchain technology. With the increasing usage of Internet of Healthcare Things (IoHT) devices, centrally managing them and ensuring the privacy of healthcare data is becoming increasingly difficult. A shard-based blockchain architecture is suggested for federated learning networks to enhance scalability and privacy, offering improved administration and privacy controls for data sharing within the network.

We discussed an innovative and reliable access control mechanism for sharing healthcare-oriented FL models that utilize permissioned blockchain technology to solve access control and privacy concerns. A distributed federated learning-based model sharing architecture designed to safeguard the privacy and security of healthcare data through the implementation of an Attribute-based Access Control (ABAC) model, whereby the trust attribute is computed and used for access control decisions.

Blockchain, as a type of distributed ledger technology, is inherently trustworthy; however, it lacks computing capability and has excessive delay due to its cumbersome consensus methods. We explore an alternative scaling strategy for a distributed healthcare federated learning-based secure model sharing architecture. The system employs state channels to minimize on-chain transactions, mitigate architectural delay, and decrease bandwidth use, therefore easing the strain on the blockchain.

For a deeper understanding, we evaluate two layer-2 blockchain methodologies (ZK-Rollup and State Channels) on throughput, latency, complexity, and manageability to enhance scalability and privacy in healthcare data exchange. We found that state-channels become more complex to manage with the increasing number of participants in the consortium network and adopted the ZK-Rollup L-2 solution for further evaluation. To improve the performance of the ZK-Rollup L-2 method, we introduce a novel concept of Proof of Availability (PoA) for the scalability of ZKP and the privacy of healthcare data. Commitment values are calculated over Electronic Health Records (EHR) using Kate–Zaverucha–Goldberg (KZG) polynomial and ZK-Succinct Non-Interactive Argument

of Knowledge (SNARK)-based ZKP system (PLONK) is used to generate proofs with these commitment values. Instead of raw data, it significantly reduces the proving time in the ZK-Rollup system. This novel system not only provides the proof of availability of data but also correctness, data retrievability, consistency with the commitment values, and ensure maximum privacy of healthcare data with lower overhead on the ZK-Rollup system.

Key words: Permissioned Blockchain; Healthcare; Security; Privacy; Scalability; Sharding; Federated Learning (FL); Hyperledger Fabric; State Channels; Zero-Knowledge Proofs (ZKP); ZK-Rollup; PLONK; Groth16; ZK-DA (Data Availability).

Scientific field of the dissertation (SSD):

Blockchain and DLT: Healthcare and Well being

School the Ph.D. Student belongs to at UNICAM:

School of Advanced Studies

University where the Ph.D. student carried out his/her program:

University of Milan

Acknowledgments

I would like to express my sincere gratitude to my primary supervisor, **Dr. Stelvio Cimato**, for their invaluable guidance, academic insight, and continuous support throughout the course of this doctoral research. Their expertise, thoughtful supervision, and constructive feedback have been essential in shaping both the direction and quality of this thesis. I am deeply appreciative of their commitment and encouragement during all stages of my doctoral studies.

My appreciation is also extended to the PhD Coordinator, **Prof. Flavio Corradini**, for their administrative guidance, support, and assistance throughout the doctoral program. Their efforts in facilitating a structured and supportive research environment are sincerely appreciated.

I would also like to extend my sincere thanks to my external supervisor, **Dr. Muslim Jameel Syed**, for their valuable suggestions, critical perspective, and technical expertise, which significantly contributed to the refinement and broader relevance of this research. Their support and collaboration are gratefully acknowledged.

I am profoundly grateful to my parents for their constant support, encouragement, and confidence in me throughout my academic journey. Their values, patience, and sacrifices have provided a strong foundation for my personal and professional development.

Above all, I wish to express my deepest gratitude to my wife for her unwavering support, understanding, and patience throughout the course of my doctoral studies. Her encouragement, strength, and steadfast belief in me were a constant source of motivation, particularly during challenging periods of this research. Her support was indispensable to the successful completion of this thesis.

Contents

List of Tables

List of Figures

List of Acronyms

1DCNN One-Dimensional Convolutional Neural Network)

ABAC Attribute-based Access Control

ACL Access Control List

BCFL Blockchain Federated Learning

CPU Central Processing Unit

DA Data Availability

DLT Distributed Ledger Technology

DoS Denial of Service

DP Differential Privacy

ECG Electrocardiogram

EHR Electronic Health Records

FL Federated Learning

GDPR General Data Protection Regulation

GPU Graphics Processing Unit

gRPC gRemote Procedure Calls

GRU Gated Recurrent Unit

HIPAA Health Insurance Portability and Accountability Act

HLF Hyper Ledger Fabric

HTTP Hyper Text Transfer Protocol

IBFT Istanbul Byzantine Fault Tolerance

IIoT Industrial Internet of Things

IMM Identity Management Module

IoV Internet of Vehicle

IPFS Inter-Planetary File System

IVC Incremental Verifiable Computation

KZG Kate–Zaverucha–Goldberg

L-2 Layer-2

MEV Maximal Extractable Value

ML Machine Learning

MRI Magnetic Resonance Imaging

NIST National Institute of Standards and Technology

NIZKP Non-Interactive Zero-Knowledge Proof

OCR Office of Civil Rights

PBFT Practical Byzantine Fault Tolerance

PDM Policy Decision Module

PLONK Permutations over Lagrange-bases for Oecumenical Non-interactive Knowledge

PoA Proof of Availability

PoFL Proof of Federated Learning

PoS Proof of Stake

PoT Powers of Tau

PoW Proof of Work

R1CS Randk-1 Constraint System

RBAC Role-based Access Control

SCs Smart Contracts

SGD Stochastic Gradient Descent

SNARK Succinct Non-Interactive Argument of Knowledge

SPoF Single Point of Failure

SRS Structured Reference String

SSD Solid State Drive

STARK Scalable Transparent Argument of Knowledge

SV Shapley Value

TMM Trust Management Module

TPS Transactions per Second

WBAN Wireless Body Area Network

ZK-DA Zero-Knowledge Data Availability

ZKP Zero-Knowledge Proof

ZKVM Zero Knowledge Virtual Machine

Chapter 1

Introduction

Blockchain has become one of the most innovative and promising digital technologies in recent years. Although designed as a basis for cryptocurrencies like Bitcoin and Ethereum, it has already shown powers beyond finance. It is one of the most promising information management solutions due to its decentralized architecture and ability to assure data integrity without a central authority.

Blockchain security is a strength, but new threats and flaws appear quickly. Blockchain consensus mechanisms, public or private, take time to validate transactions. It performs poorly with high transaction volumes. Users wait for transaction validation, sometimes for long periods. This thesis proposes a Layer-2 (L2) methods like State Channels and ZK-Rollups to improve blockchain network scalability. Healthcare data security and privacy are the focus of this dissertation. The growing number of digital healthcare data and medical records makes privacy and compliance with strict data privacy requirements difficult. Scalability is another issue with growing records. This thesis aims to use blockchain technology to preserve healthcare data privacy and scalability.

1.1 Overview

1.1.1 Overview of the Topic

In recent years, blockchain has emerged as one of the most inventive and promising technologies in the global digital arena. Initially conceived as a foundational framework for cryptocurrencies like Bitcoin and Ethereum, it has rapidly revealed capabilities beyond the financial domain [87].

Its decentralized architecture and the capacity to ensure data integrity without a central authority make it one of the most promising alternatives for the future of information management.

The essence of blockchain is a consensus mechanism, a decentralized management framework enabling independent nodes to authenticate and disseminate transactions inside blocks. This method improves efficiency and offers nearly inherent protection against assaults and manipulation efforts [46].

Although blockchain security is a notable asset, it does not provide total protection;

new threats and weaknesses arise rapidly. A major factor is inefficiency of time. The consensus methods used by any blockchain system, whether public or private, need significant time to validate transactions. It fails to yield satisfactory results when the volume of transactions is elevated. Users must wait, occasionally for extended durations to validate their transactions. This study aims to provide an efficient framework that enhances scalability in the blockchain network using a Layer-2 (L-2) approach [72].

This study subject primarily emphasizes the importance of healthcare data security and privacy. The increasing volume of digital healthcare data and medical records presents challenges in maintaining privacy and adhering to stringent data privacy regulations. Scalability is an additional problem in managing the increasing volume of records. The primary focus of this research is to introduce an innovative method to protect the privacy of healthcare data while ensuring scalability through the implementation of blockchain technology.

1.2 Blockchain and DLT

Bitcoin was introduced in 2009 as a peer-to-peer electronic currency. Bitcoin catalyzed the emergence of other alternative cryptocurrencies [19]. Bitcoin established the foundation for a novel domain known as blockchain, which is a specific form of distributed ledger technology (DLT) [84]. DLT is a comprehensive phrase that denotes any digital framework for concurrently documenting asset transactions in various locations. Data are distributed and duplicated between nodes without a singular central authority. The blockchain is a subset of DLT in which records are organized in blocks that are sequentially connected in a chain. The blockchain is a linear data structure that includes a sequence of blocks, each containing a collection of transactions. The longest chain is considered the valid chain. Examples of blockchains include Bitcoin and Ethereum [87].

Although the concept is straightforward, blockchain technology is exceedingly complex. Even original blockchains, such as Bitcoin, are multilayered systems that employ a series of algorithms to ensure the authenticity and immutability of the information within the ledgers. Smart contracts extend capabilities by enabling programs to operate directly on the blockchain.

Before examining security and privacy considerations, as well as potential vulnerabilities of the system, it is essential to understand its architecture and the underlying technological principles.

1.2.1 Objectives of Blockchain

Blockchain seeks to offer a substitute for the management of a digital ledger [41]. The design provides certain features that must be fulfilled, serving as a representation of the achievement of defined objectives.

Decentralization

A primary purpose is the establishment and maintenance of a digital ledger, disseminated among several organizations known as nodes, without a central authority capable of making choices for all. Historically, maintaining of a ledger required a central authority responsible for decision-making on behalf of all participants. Blockchain systems employ consensus procedures, cryptographic primitives, and many tools to supplant central authority; however, many do not achieve complete decentralization as planned. Consensus algorithms often promote centralized control as time progresses. For example, in Proof of Work (PoW), miners often create alliances to obtain more rewards, but in Proof of Stake (PoS) [37], individuals with greater financial assets typically maintain control over the blockchain for extended periods.

Immutability

As each node is tasked with preserving its own version of the ledger, a method is required to inhibit unauthorized modifications or augmentations by the nodes. Hash functions, digital signatures, and cryptographic methods are used to prevent the generation of counterfeit blocks and the alteration of information, which may otherwise be validated by the network.

Anonymity

This technology's significant promise lies in its architecture, which associates asset ownership or transaction origination with account addresses instead than actual individuals. Account addresses are derived from a randomly generated pair of public and private keys, which should ensure a substantial level of anonymity.

In the real world, this system offers pseudo-anonymity rather than total anonymity [41], as it aggregates publicly accessible information from the ledger transmitted across all nodes, enabling the identification of an account owner, and so undermining this promise.

Transparency

Centralized information management solutions are typically opaque. For example, data held by healthcare organizations is not publicly accessible, complicating the assessment of management practices within a chain. Therefore, data owners (patients) must have confidence that healthcare organizations will operate justly and prioritize their data privacy. Conversely, Blockchain employs a transparent digital ledger. All nodes within the network are tasked with replicating and verifying all blocks prior to their incorporation [104].

1.3 Research Objectives

The first objective of this research is to elucidate blockchain and Distributed Ledger Technology (DLT) to facilitate understanding of their operational principles and to explore

their application in improving the security, privacy and scalability of digital healthcare data.

The second objective is to explore the blockchain technology for the establishment of a consortium blockchain. How a consortium or permissioned blockchain improves management controls, access control, and privacy for healthcare data stored and processed at many healthcare institutions

The third objective is to design, implement, and evaluate a secure and privacy-preserving framework for sharing healthcare data in distributed environments. This objective focuses on investigating how permissioned blockchain architectures can support trustworthy data exchange among heterogeneous healthcare stakeholders while enforcing fine-grained access control, accountability, and compliance requirements. The research explores the integration of smart-contract-based governance mechanisms to regulate data access and usage, as well as the development of monitoring and detection mechanisms to identify unauthorized access, misuse, or anomalous behavior within the healthcare data sharing ecosystem.

The fourth objective is to employ scalable solutions for blockchain technologies and healthcare big data that can reduce the overhead of blockchain transaction validations and healthcare data exchange. Different Layer-2 or off-chain solutions are discussed to find the best approach to provide scalability to healthcare data.

The fifth objective is to introduce a novel cryptographic approach to improve the privacy and integrity of blockchain technology for healthcare data. Blockchain technology with this cryptographic approach can strengthen the overall healthcare data exchange architecture. In addition, the approach should be scalable and provide significant results with increasing data record sizes and numbers of transactions.

1.4 Research Questions

There are different research questions, which are addressed here briefly.

- RQ1: Privacy Preservation: Ensuring privacy in the healthcare data management is a critical challenge. How can healthcare data management systems be designed to achieve strong data privacy with cryptographic primitives and a decentralized system like blockchain?

  To preserve the privacy of healthcare data, blockchain technology is used to provide a privacy preserving framework with identity verification models, fine-grained access management, and little or no exposure of healthcare data functions.

- RQ2: Scalability: With the growing number of healthcare digital records, how can we provide scalability to decentralized healthcare data systems to manage large-volume data exchange and efficient computational resources while preserving privacy. Which scalable methods facilitate high-throughput and low-latency healthcare data processing with blockchain and distributed ledger systems?

In blockchain, Healthcare data workloads are different from financial workloads. Healthcare data has large volumes. For this purpose, off-chain solutions can be integrated with the blockchain to provide scalability for both healthcare data and the blockchain network itself, allowing most operations to be performed without interacting with the blockchain network. Exploring these off-chain solutions helps to find a high-throughput mechanism for healthcare data processing.

- RQ3: Interoperability and Architecture: How can cryptographic primitives can be integrated with the healthcare data systems to achieve privacy-preserving, correctness and scalable interoperability across heterogeneous healthcare institutions while maintaining regulatory compliance?

  Interoperability is difficult in healthcare due to heterogeneous data formats, regulatory limitations, and legacy systems. Therefore, these healthcare data systems require blockchain coordination by providing a gateway to exchange healthcare data among heterogeneous healthcare institutions.

- RQ4: Performance Evaluation: How does the integration of cryptographic primitives and scalable blockchain approaches provide better performance in the throughput and latency of healthcare data exchange operations while ensuring privacy? What are the quantitative improvements in latency and throughput when using ZKP systems with the Layer-2 blockchain?

  Performance evaluation is about to define performance metrics such as throughput and latency. Performance evaluation is important because healthcare systems manage and store large amounts of healthcare data and engage in frequent data access. For addressing scalability, throughput, latency, and efficiency metrics are required to be presented with scaling curves. In terms of privacy, it is essential to share how much privacy is achieved by using privacy-preserving methods.

- RQ5: Access Control: How can an access control mechanism provide better access management to healthcare data requests? How do blockchain's smart contracts verify requests from different healthcare institutions to provide secure access to healthcare data? What access control model suits well the healthcare data architecture?

  Different access control mechanisms and models can be integrated with blockchain frameworks. Several blockchain frameworks have built-in access control mechanisms to provide data access management; others provide flexibility to integrate advanced access control models like Attribute-based Access Control (ABAC) for more fine-grained access management. Access policies can be deployed on the blockchain in the form of smart contracts, which are difficult to change without consensus approval in a consortium network.

1.4.1 Key Contributions

The use of machine learning (ML) techniques in electronic health records (EHR) is increasingly recognized as a means to extract significant information that may improve decision-making in healthcare. An effective method originates from the federated learning (FL) paradigm, which enables decentralized training of machine learning models with locally stored datasets. It serves as a valuable instrument when used with blockchain technology. The increasing use of IoT healthcare (IoHT) devices is complicating centralized management and safeguarding healthcare data privacy. We propose an architecture to handle the scalability issue of healthcare data management in federated learning networks utilizing a sharding-based blockchain method [73].

Blockchain shards provide scalability on-chain, mostly healthcare data and trained models are exchanged among healthcare organizations more frequently. Therefore, it is important to address privacy and scalability off-chain. State channels provide an off-chain scalability solution, where data is exchanged by creating a dedicated channel between two participants, and data exchange states are maintained. Once the data are fully exchanged, a final state is submitted on the blockchain with the agreement of the participants [74]. The terms and conditions of this agreement are written in a smart contract deployed on the blockchain to create a specific state channel. Once the data are exchanged and participants agree on successful data exchange, the smart contract is called to submit the final state. The smart contract verifies the digital signatures of the participants to close a channel. In this way, only final states are submitted to the blockchain and not the intermediate states.

In addition with scalability, our contribution is also towards the security and access control of the healthcare data. We integrated an access control mechanism (ABAC) to manage access requests to healthcare data. We implement this ABAC model with a case scenario in an FL environment. In addition, we implemented a trust-based attribute to detect malicious nodes in the FL network and isolate nodes with lower trust values from participating in the FL training process with the help of the ABAC model.

Our major contribution is binding scalability and privacy factors to propose a unique solution for healthcare data using blockchain. We evaluated two layer-2 approaches (state channels and ZK-Rollup) to find the best approach for a robust and efficient blockchain-based scalable and privacy-orientated solution. By choosing ZK-Rollup as a more appropriate layer-2 solution for healthcare data architecture, we contributed a novel solution of Data Availability (DA) with PoA that not only provides strong privacy for healthcare data, but also makes it scalable and faster to use. The proof of availability (PoA) ensures data availability, retrievability, and correctness and generates a proof for final validation on the blockchain.

1.5 Structure of the Thesis

This thesis is organized into eight main chapters, each focusing on a key component of Blockchain privacy and scalability in relation to healthcare data.

In the first chapter, we present an overview of the thesis topic and the overall objective to achieve in this research work. We introduce the fundamentals of blockchain, analyze its structure, and the main security and privacy challenges. The chapter presents the research questions that are discussed in subsequent chapters.

The second chapter delineates the structure of blockchain technology and its operational parameters. This thesis focuses primarily on the permissioned blockchain, and its structure is examined in this chapter.

In the third chapter, we start our discussion with a case study, where a federated learning (FL) network trains models on the healthcare data, and we propose the sharding method of permissioned blockchain (Hyperledger Fabric) for the scalability of the FL network.

In the fourth chapter, we introduce the concept of state channel integration with permissioned blockchain, which is a L-2 approach to the blockchain scalability. We develop a scenario of sharing model updates trained in a federated learning network over a state channel. The general concept is to reduce the on-chain interaction and perform model update tasks off-chain. In this case scenario, different healthcare entities (hospitals, medical laboratories, research centers, etc.) train healthcare models in their local network, and once the training process completes, then only the final state of these models uploads on the blockchain.

In the fifth chapter, we present the concept of ABAC model integration with blockchain. In a permissioned blockchain, identity and access management is an intrinsic feature that works with Role-based Access Control (RBAC) model. To improve the privacy and access control of healthcare data, we integrate the ABAC model with blockchain for fine-grained access to healthcare data. We create a case scenario with a federated learning network where model sharing requests are managed by the proposed ABAC model. In addition, we create a Trust Management Module (TMM) to calculate the trust values of the participating nodes in the federated learning network and detect malicious nodes based on these trust values.

In the sixth chapter, we start our discussion with the introduction of Zero-Knowledge Proof (ZKP), which is also a core topic of this topic to ensure the privacy of healthcare data. We discuss different types of ZKP systems and their implementation in different setups. This chapter also provides the selection criteria for the ZKP system for our research problem. We explore ZK-Rollup technology, which is another L-2 approach to provide scalability to the blockchain network. We compare ZK-Rollup and state channels to find the best approach to the scalability of blockchain and the privacy of healthcare data.

In the seventh and last chapter of this thesis, we further explore the ZK-Rollup system and present a novel solution to enhance the performance of ZK-Rollup to provide

scalability and privacy to healthcare data. We introduce a concept of PoA that ensures the availability of data in correct and complete form by reducing the computational overhead for proof generation.

1.6 Conclusion

This chapter initiates our formal discourse on blockchain and DLT. We examined a fundamental distinction between blockchain and DLT with respect to blockchain aims. We enumerated the aims of this research and thoroughly answered the research questions pertaining to the topic. In the preceding part, we delineated the structure of this thesis.

Chapter 2

Structure of Blockchain

This chapter provides details about the working structure of the blockchain technology. We discuss the types (public and private) of blockchain and their working and their way of working. Additionally, we discuss the blockchain layers with their functionalities. In this dissertation, our focus is on the private or permissioned blockchain; therefore, we also discuss the structure and working structure of permissioned blockchain in this chapter.

2.1 Introduction

Blockchain is presented as a transformative and distinctive technology for implementing distributed ledgers, attracting significant research interest in both academia and business. The initial introduction of blockchain was by Satoshi Nakamoto, who created Bitcoin as a cryptocurrency to establish a reliable peer-to-peer network to conduct transactions [54].

A blockchain network consists of a sequence of blocks that include transactional records maintained by distributed ledgers, such as public ledgers. In recent years, digital ledgers have been utilized for centralized data storage; however, blockchain technology offers a distributed and decentralized method of recording data. The initial block in a series of blocks is called the genesis block. Beginning with the second block, each subsequent block has the hash value of its predecessor, therefore establishing the sequence of blocks.

Each block comprises two components: block data, which contains transaction records, and the block header [94]. The header of the block is divided into the following components:

- Block version: It indicates the version of the block format and protocol used. Typically,it has a size of 4 bytes or 32 bits.
- Merkle root hash: A Merkle tree is used to compute the hashes of all the transactions in the block.
- Previous block hash: This part holds the hash value of the previous block and typically has a size of 32 bytes.
- Timestamps: This value indicates the time when a block was created and maintains the chronological order. This value is used in consensus and validation rules. The

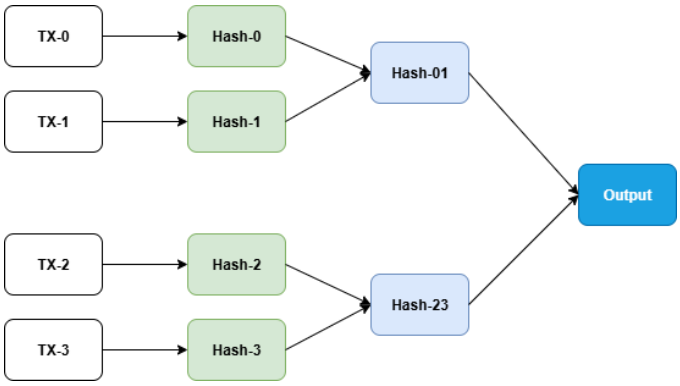

Figure (2.1) Merkle Tree Root Hash Function

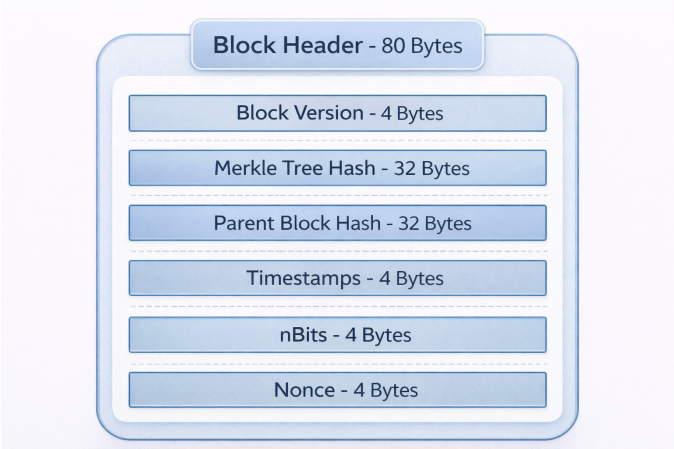

Figure (2.2) Block Header

size of the timestamp value is 4 bytes.

- nBits: This value is required for block validation and it defines how hard it is to produce a valid block. It is also 4 bytes in size
- Nonce: It presents a variable number that miners change to find a valid block hash. This value is used in Proof of Work (PoW) to satisfy difficult conditions.

The block data comprises transactions and transaction identifiers. Each block stores encrypted version of transactions. The hash values of these transactions are generated. Subsequently, each two-hash value of every transaction is paired and subjected to an additional hash algorithm to generate a new hash value. This procedure is called the Merkle tree root hash and is shown in Figure 2.1.

A pictorial representation of the header and structure of a blockchain block is shown in Figure 2.2 and Figure 2.3

2.2 How Blockchain Works?

Users using blockchain technology are called nodes. As the blockchain functions as a distributed ledger, each node within the network possesses a complete copy of the entire

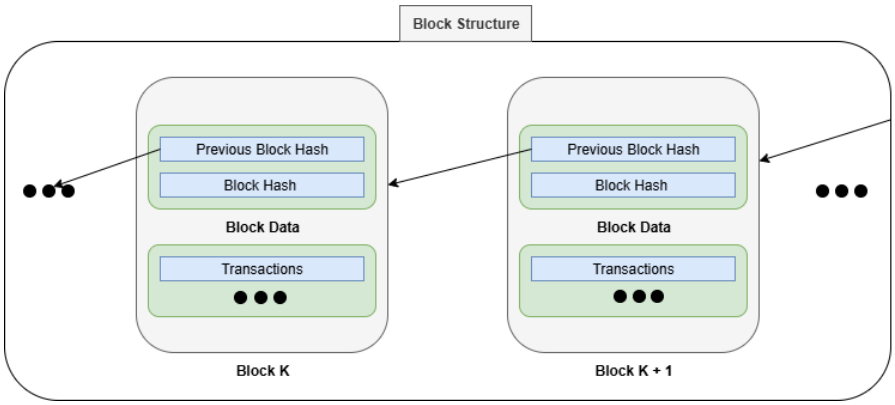

Figure (2.3) Block Structure

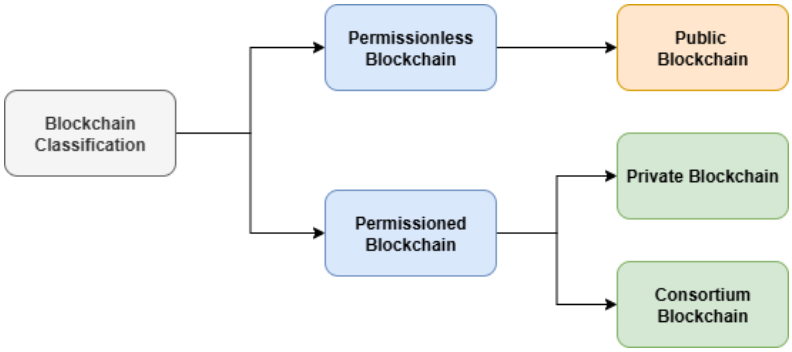

Figure (2.4) Blockchain Classification

blockchain [80]. Certain nodes possess special features that enable them to add a new block to block sequences. These nodes function as miners and execute the transaction validation process.

Each node can initiate a transaction by employing a digital signature utilizing public and private key cryptography. Each node possesses the public keys of other nodes, whereas each private key is exclusively known to its owner.

Transactions are sent to all nodes using the Gossip protocol. Subsequently, the miners verify and validate the transactions, and a verified transaction is added to the block. This procedure is called mining.

2.3 Classification of Blockchain

Blockchain networks are categorized into two types based on the authorization to add a new block. If any user may access and create a new block, the blockchain is considered permissionless. However, if only a limited number of designated users are authorized to contribute a new block to the network, the blockchain is considered permissioned. A permissionless blockchain is similar to the public Internet, accessible to anyone, but a permissioned blockchain is like an Intranet, which is regulated and controlled. A permissioned blockchain is frequently used by collectives or organizations [86].

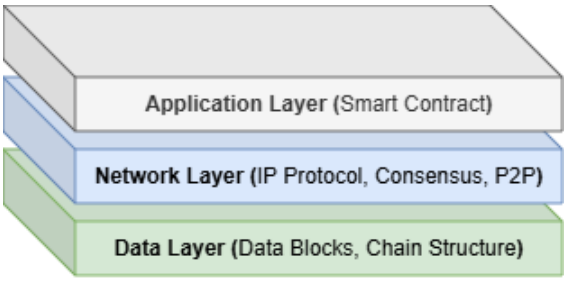

Figure (2.5) Blockchain Layers

2.3.1 Permissioned Blockchain

Permissioned blockchain is divided into two primary categories: private blockchain and consortium blockchain. In the first group, only a subset of nodes is permitted to access the network and authenticate the transaction initiated by any permitted node. This constraint improves transaction speed, ensuring efficiency even as the blockchain expands. This form of blockchain enables for entirely safe transactions. The primary drawback of the private blockchain is that its ownership by a single entity results in a centralized validation process [4].

A consortium blockchain possesses characteristics of both public and private blockchains, in which a trusted group of organizations governs the blockchain network. Members of this group is called nodes that validate transactions and maintain consensus. Only authorized members of the group can join the consortium blockchain network. [24].

2.3.2 Permissionless Blockchain

A prime illustration of permissionless blockchain is the public blockchain. In a public blockchain, any user can join or leave the network at any time [91]. All users own a singular copy of all transactions and due to the transparency of these transactions, no individual may alter them. The most renowned public blockchain is Bitcoin [47].

2.4 Blockchain Protocol Layers

The blockchain structure has three layers: the network layer, the data layer, and the application layer, as illustrated in Figure 2.5. The data layer is responsible for constructing new blocks, the network layer governs the interaction protocols between users and the environment, and the application layer represents the diverse applications of this technology.

2.4.1 Data Layer

The Data Layer is the fundamental basis of a blockchain. The data layer offers the functional and procedural mechanisms for creating data entries (blocks) in the blockchain ledger. This layer pertains to data structures and methods, including digital signa-

tures, Merkle trees, timestamps, and hash pointers. It makes the blockchain network decentralized, transparent, and durable.

2.4.2 Network Layer

The network layer facilitates decentralization of the network using IP protocols and peer-to-peer networks. This layer delineates the consensus algorithm and its implementation utilized for distributed agreement in block validation.

2.4.3 Application Layer

The application layer constitutes the highest layer of the blockchain structure. This layer shows the diverse applications of blockchain across several areas to utilize its distributed and decentralized features. Integrates blockchain capabilities with practical applications, allowing people, companies, and other systems to engage with the network through smart contracts, APIs, and decentralized apps (DApps).

2.5 Structure of Permissioned Blockchain

A permissioned blockchain is a sort of distributed ledger in which only authorized participants may access the network, make transactions, validate blocks, and retrieve data. In the healthcare context, hospitals, medical laboratories, research institutions, insurers, and regulators/auditors are the pertinent entities to participate in the permissioned blockchain network. In this thesis, we are mainly working with permissioned blockchain for the healthcare data privacy and scalability, therefore, it is important to understand the structure of permissioned blockchain specifically. We discuss the structure of permissioned blockchain with respect to its [21]. In Figure2.6 a conceptual layered structure of permissioned blockchain is presented.

2.5.1 Application Layer

This layer provides a main interaction point for users by using web applications, an API (Application Programming Interface), and a dashboard. The applications for sharing records by the patients and the insurance claim system used by insurance companies are examples of the application layer. Smart contracts (chaincodes) are also deployed in this layer, which encodes business rules, validates transactions, and enforces access control policies.

2.5.2 Transaction and Endorsement Layer/Network Layer

This layer collects transactions from the application layer and endorses these transactions to the peer nodes based on policies set in chaincodes. Not every peer node supports the transaction. The purpose is to improve the scalability of the system. For example,

identity management smart contract to manage authorized users and business logic smart contracts to run blockchain operations.

2.5.3 Ordering Layer

It is also called the consensus layer. Ordering nodes endorsed the transaction into blocks and ensures global agreement on transaction order. In permissioned blockchain, there are commonly two consensus mechanisms used (Practical Byzantine Fault Tolerance (PBFT)/Istanbul Byzantine Fault Tolerance (IBFT), Raft). However, in this thesis, we are using the Raft consensus algorithm in our permissioned blockchain setup. Provides fast finality and lower energy consumption while ordering and creating blocks [26].

2.5.4 Identity and Access Management

This is an extensive layer in the permissioned blockchain; it authenticates participants, issues certificates, and provides access control. By default, it creates PKI (Public Key Infrastructure) [27] and provides RBAC (Role-Based Access Control) [15]. We implemented ABAC (Attribute-based Access Control) for the blockchain network that we will discuss in the next chapter. In the healthcare context, only authorized doctors can access healthcare data; patients can control their access rights.

2.6 Raft Consensus Algorithm

By focusing on majority-based log replication rather than multi-round voting, Raft is a leader-based crash fault tolerant consensus protocol that delivers high throughput and low latency. Raft is a protocol designed to be crash-resistant [30]. Raft grows more effectively with the number of nodes, giving O(n) communication, simpler implementation, and much greater performance than PBFT, which tolerates Byzantine errors but incurs O(n2) message complexity. The Raft also offers significantly superior performance. Raft is a favored consensus mechanism over PBFT because it offers an effective balance between scalability, dependability, and operational simplicity [38]. This makes it an ideal choice for permissioned blockchain, such as healthcare consortia, where members are authorized and governed.

2.7 Conclusion

In this chapter, we discussed the working structure of public and private blockchain networks. How transactions are managed with hash functions and what information is stored in the blocks. Moreover, we discussed the blockchain layers and their functionalities. Our focus in this thesis is on the permissioned blockchain; therefore, it is important to discuss the permissioned blockchain structure. How is it different from the public blockchain and what is the layered structure. In a permissioned blockchain, Raft is a very

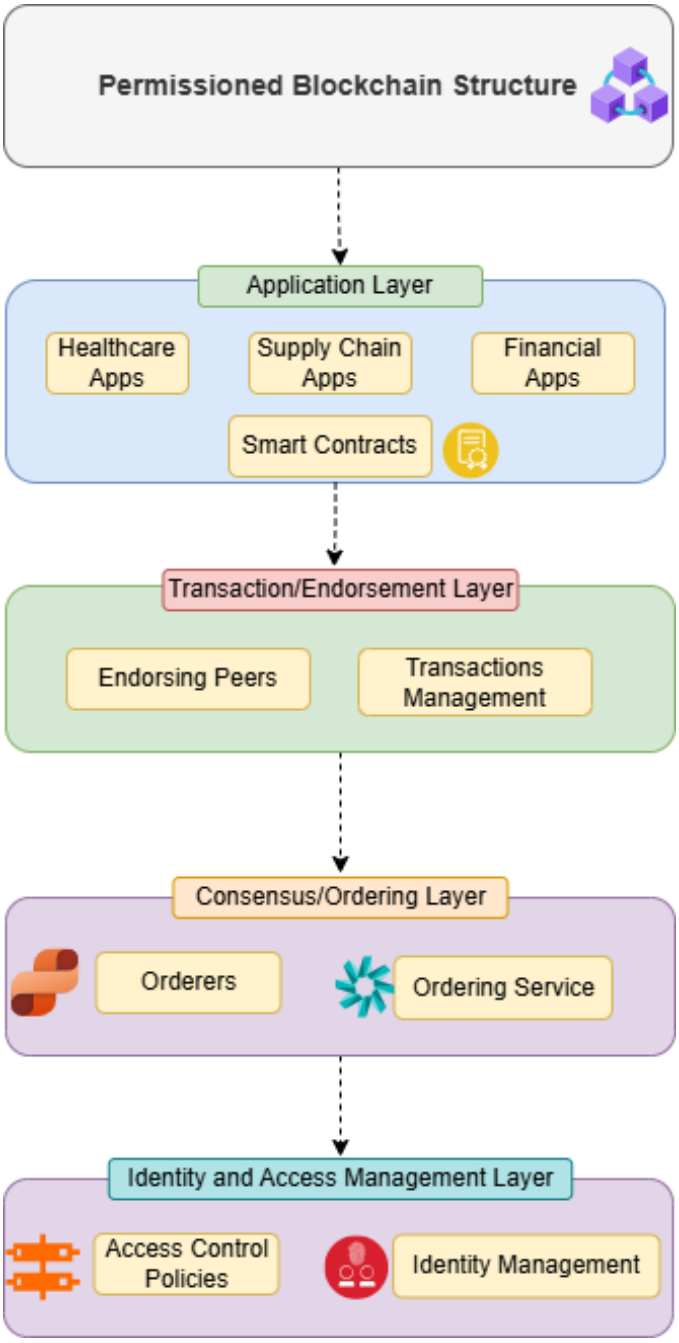

Figure (2.6) Permissioned Blockchain Layers

popular and fast consensus algorithm, mostly used in the permissioned blockchain, such as Hyperledger Fabric [85].

Chapter 3

Federated Learning and Blockchain

In this chapter, we discuss a permissioned blockchain-based architecture for healthcare data which uses sharding to address scalability issues in a federated learning (FL) network, used to train model on distributed IoHT devices. The scalability issue raises when a large number of healthcare devices are used to train the Machine Learning (ML) models by using the FL technique.

3.1 Introduction

Federated Learning (FL) is a new category of distributed machine learning methods in which the training process is performed without consolidating data in cloud data centers [36]. The execution of a typical federated learning process involves the interaction between centralized Internet-enabled servers and the distributed devices and systems that are connected to them over the Internet [7]. The use of the model-first approach in FL results in a reduction of data communication costs. The FL facilitates centralized servers to maintain a comprehensive model and disseminate the model parameters to interconnected devices and systems, rather than retrieving extensive data sets [36]. FL is a distributed machine learning protocol that prioritizes privacy preservation. In this scenario, devices start the training process using the global model provided by the server. Subsequently, they train local models using their respective datasets [17].

Although FL exhibits enhanced performance and efficacy in privacy preservation, optimum bandwidth, and reduced latency, it is still constrained by some security and privacy limitations. Each iteration of the federated learning model update procedure culminates in total reliance on the central federated learning server. The failure of a central server creates a Single Point of Failure (SPoF) scenario, which can result in Denial of Service (DoS) attacks [61].

In this chapter, we are introducing the concept of sharding, which is a scalability technique used in blockchain networks, especially in private blockchains, where a single network is divided into multiple networks or groups called shards. Each network or group

manages or processes a subset of blockchain transactions and smart contracts, enabling the network to perform operations in parallel but not sequentially. In a traditional blockchain network, every node processes every transaction, which limits scalability, but with the sharding technique, the overall network is divided into multiple shards, and each shard maintains its own state and transactions. Additionally, we can do cross-sharding that enables communication and transaction execution between different shards. It can provide better access control of data flowing from one network to the other. In this chapter, we introduce the concept of sharding, which is a scalability technique used in blockchain networks, especially in private blockchains, where a single network is divided into multiple networks or groups called shards. Each network or group manages or processes a subset of blockchain transactions and smart contracts, enabling the network to perform operations in parallel but not sequentially. In a traditional blockchain network, every node processes every transaction, which limits scalability, but with the sharding technique, the overall network is divided into multiple shards, and each shard maintains its own state and transactions. Additionally, we can do cross-sharding that enables communication and transaction execution between different shards. It can provide better access control of data flowing from one network to the other.

It is better to discuss state channels to get a general idea of how they work. State channels are off-chain communication mechanisms that enable authorized participants to perform multiple transactions off-chain or privately. For example, if two participants are required to execute a number of transactions, then in the case of blockchain, they need to upload the state of each transaction to the main blockchain. To validate a single transaction, the main blockchain will take time due to its consensus process. The subsequent transactions need to wait until the currently uploaded transaction is validated on the main blockchain. It creates a scalability issue; the state channels enable participants to create a private channel among them and execute as many transactions as they want to do, and only the final state or a summary of all the transactions is uploaded on the main blockchain. It provides scalability, speed, and throughput to the blockchain network.

Several other security issues are associated with the FL. These issues include model inversion attacks, model poisoning attacks, inference attacks, data label poisoning attacks, and authentication and permission requirements for devices. With the help of smart contracts that ensure cross-shard communication, security can be provided to FL tasks, including model parameter sharing, local model generation, and global model storage. More crucially, it replaces the central coordinator's function with a blockchain that supports a decentralized, federated learning environment [40]. We discuss a proof of concept implementation of the proposed architecture deploying Hyperledger framework, and reporting some performance analysis we conducted to analyze the transactions throughput in relation to the number of shards.

3.1.1 Blockchain for Federated Learning

The above stated attacks can be mitigated with the implementation of blockchain technology in federated learning systems. Blockchain technology can solve these challenges by enabling decentralized storage of model updates and ensuring traceability of the updated model. In addition, blockchain technology uses a combination of chain, tree, and graph structures to guarantee immutability and preserve a complete record of past data. Furthermore, addressing the challenge of large data in healthcare care with federated learning has several benefits, including preservation of privacy, scalability, decreased data transmission, and resistance to network failures.

In this chapter, we discuss a blockchain-based architecture for healthcare data which uses sharding to address scalability issues, raised when a large number of healthcare devices are used to train the ML models by using the FL technique.

3.1.2 Sharding in Blockchain

The throughput of a blockchain system, quantified as the number of transactions processed per second, significantly falls short of actual needs, hence becoming a critical restriction that hinders widespread adoption of blockchain technology. For example, Bitcoin is capable of processing only about 10 transactions per second due to its maximum block size of 1 MB and an average block interval of 10 minutes. This significantly obstructs the application of blockchains in high-frequency transactions. The blockchain is regarded as a secure foundation layer or settlement center for cryptocurrencies, facilitating the processing of numerous transactions off-chain before their final settlement on the blockchain. The Lightning Network and Raiden Network, which utilize state-channel technology, provides off-chain payments and forwards a summary of a batch of these payments to the blockchain. These techniques, referred to as Layer-2 scaling, minimize contact with the Blockchain to alleviate latency from the user's viewpoint.

In contrast, layer-1 scaling is intended to enhance the throughput of blockchains from a systemic point of view. A blockchain system may be enhanced in the following manners to accommodate an increasing workload.

- Minimizing communication and computational overhead
- Adding resources to a single node, i.e., vertical scalability;
- Incorporating more nodes into the Blockchain, namely horizontal scalability.

Minimizing Overhead: New blockchain consensus methods have been created to improve blockchain performance by minimizing overhead. The conventional PBFT consensus mechanism has been refined to minimize communication overhead and achieve high throughput in large networks.

Vertical Scaling: To enhance throughput using vertical scaling approaches, the number of authorized transactions is raised in a single block, and reducing block time can improve blockchain throughput but require more resources, for example, storage,

computation, and bandwidth. Nevertheless, vertical scaling methods cannot indefinitely enhance throughput, as a Blockchain system is structured to operate inside a decentralized and homogeneous network, where security is tightly linked to consensus throughout the whole network. The larger the network, the greater the bandwidth required for synchronization.

Horizontal Scaling/Sharding: Sharding technology, which segments a whole blockchain into many sub-blockchains or shards, enables participating nodes to execute and store transactions from selected shards (subpart of the blockchain), thus facilitating horizontal scaling. Using sharding technology, which enables partial transaction processing and storage on individual nodes, the whole Blockchain may attain a linearly rising throughput as the number of nodes expands. The use of blockchains is crucial for delivering a substantial quantity and quality of services to the users inside expansive networks characterized by limitless expansion, hence attracting the attention of researchers focused on enhancing Blockchain scalability.

3.1.3 Sharding for Federated Learning

We propose a blockchain-based architecture for healthcare data that uses sharding to mitigate scalability challenges arising from numerous devices that generate data to train machine learning models using the FL technique.

Smart contracts facilitating cross-shard communication might enhance security for federated learning activities, encompassing model parameter exchange, local model development, and global model storage. Fundamentally, it replaces the central coordinator role with a blockchain that facilitates a decentralized federated learning ecosystem.

This method is primarily based on the use of a sharded blockchain within the Internet of Healthcare Things (IoHT) network. Designed to provide federated learning across several network segments designed to train models on diverse medical datasets. Since each shard operates as an independent element inside a cohesive global blockchain network, it is feasible to achieve both scalability and autonomy concurrently. These fragments function as elements of a cohesive global blockchain network, allowing distinct management of consensus and transactions. A global consensus can be achieved by using network connections. Each local shard consists of several nodes within a blockchain system and operates as a self-sufficient and independent network within the broader blockchain architecture.

This approach involves the use of an individual blockchain shard for each healthcare network. These networks comprise many medical organizations, each with its own localized data stored and processed by different IoHT devices. The IoHT edge devices utilize local training to create the model and then transfer the trained model to their corresponding shards. These shards function as components of a unified global blockchain network, enabling autonomous management of consensus and transactions. Each local shard operates as its own blockchain network, comprising many blockchain nodes inside it. This enables the accommodation of diverse task-related demands through the adaptation of

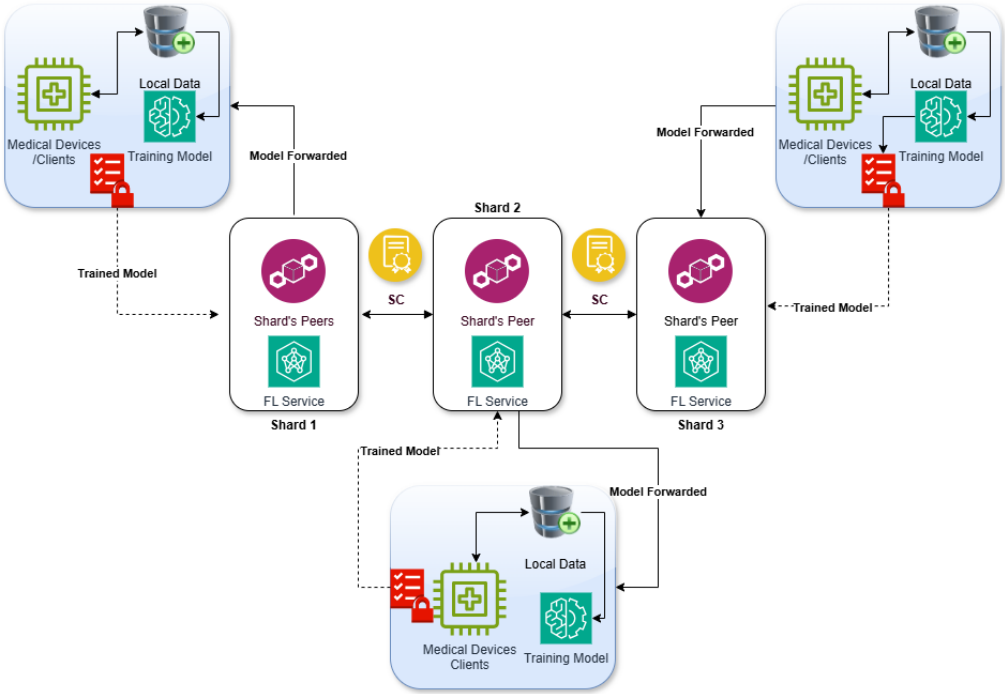

Figure (3.1) Shard-based Blockchain Architecture

consensus methods. For example, during the training of large models, when the efficacy of an individual node may be constrained, a technique like PBFT can be employed. Alternatively, shards with a reduced number of clients may utilize Raft consensus.

3.1.4 Workflow of Sharding-based Blockchain

This section thoroughly analyzes the workflow of a sharding-based blockchain, clarifying the sequential steps required to create a setup to train an FL-based model with scalability

In a traditional federated learning environment, all participating devices are considered clients. In this context, clients include IoHT devices such as wearable sensors, glucometers, blood pressure monitors, insulin pumps, and various cardiovascular equipment. Clients have the confidential healthcare datasets required for the ML model training process, and it is their responsibility to generate model updates. A comparable premise is established in which clients are perceived as medical devices that do not authenticate models but rather depend on other, more potent devices to maintain system security. Clients have different limitations on battery life, storage capacity, and computing power.

As seen in Figure 3.1, the clients or IoHT devices interact with their corresponding peers instantiated within a blockchain shard. Clients possess the healthcare data; hence, it is imperative to identify and verify legitimate clients participating in the FL process.

Blockchain-based Shards

Sharding improves scalability by dividing the network into smaller subgroups. This method decreases storage and communication requirements while enhancing throughput

as the number of shards increases. Several blockchain-based shards have been established to facilitate communication among healthcare organizations, including hospitals, pharmacies, clinics, and other medical institutions. Each shard can support several healthcare organizations, wherein numerous healthcare devices participate in the federated learning process, or designated areas where multiple healthcare organizations are established to deliver healthcare services.

In the realm of federated learning, the use of a blockchain network is viable for several parties. Each network entity possesses the ability to operate its own node within the blockchain framework. Details about peers, stringent security protocols for cross-shard communication, information regarding machine learning models, model training data, and model reports inside the federated learning environment are all preserved in the blockchain ledger storage. Consequently, the federated learning process exhibits enhanced transparency and audit ability.

Every blockchain peer offers the FL service. The implementation comprises a set of functions. The stages in the execution of these processes include the building of the federated learning pipeline, the establishment of the initial model, the production of local models by each federation member, and the synthesis of a global model through the average function.

The blockchain network may be used by several entities under a federated learning framework. Every network unit has the ability to operate its own blockchain node. The blockchain ledger maintains the peer details of the federated learning environment. The permission contract governs cross-shard transactions, machine learning model information, model training data, and model reports.

For example, smart contracts interact with the FL service to generate blocks on the blockchain and utilize the Stochastic Gradient Descent (SGD) method to calculate the average of local models and incorporate them into the global model.

Peers

The blockchain is distributed among several peers across diverse network segments, each possessing its own dataset, which includes information pertaining to IoHT devices. Federated learning can encompass several peers from various network shards. We develop a global model utilizing the updated models taught by clients and saved across many peers in the shards. Multiple peer nodes maintain copies of the blockchain, each containing a distinct set of transactions. Several peers inside each shard develop local models for training and transmit these models to the devices/clients for updates. Every peer in a shard-based blockchain possesses individual off-chain storage, wherein data hashes are locally retained and then published to the blockchain ledger. These data encompass the client's information, their trust values, and the parameters that are initiated to begin the FL procedure. For example, when identifying a malicious IoHT device, each blockchain peer possesses the device's identification information and other data kept in off-chain storage.

In the preceding paragraph, we described the attributes and characteristics of a peer. A peer has a local dataset that includes information about the clients and parameters utilized in the FL process. The peers keep a localized version of the trained model in their local blockchain ledger and aggregate it with data models from other peers to create a global model. Smart contracts can be utilized for the initialization and aggregation of the model.

Clients

In each shard, many peers are established, In previous sessions, we examined the attributes of peers. Individuals inside each shard can communicate and exchange information under the supervision of smart contracts. In this scenario, peers perform several operations: transmitting models with their parameters to clients for training, receiving updated models from clients, and certifying these updated models using a consensus mechanism. During consensus, peers are tasked with evaluating model modifications and identifying malicious clients using zero-trust security principles. Each shard may possess a varying number of peers.

3.1.5 Identification of FL Devices

In this scenario, multiple peers are coordinating with each other. Medical and healthcare organizations such as hospitals, clinics, medical laboratories, healthcare providers, pharmacies, and mobile services can act as peers. A subset of peers can also be created within a peer organization.

Each peer may have its own off-chain storage with local models and parameters used to train models from the IoHT devices. These peers must first be on-boarded with their peers' unique IDs, and all the associated medical devices must also be on-boarded with their unique device IDs to participate in the FL process. Separate blockchain nodes are created for each of these peers, and the blockchain ledger stores all of the peers and associated devices' identifying information.

A strict device verification and authorization policy is one in which no device or user is trusted unless they are authenticated. In other words, identity verification is required in a distributed system before providing any service. This guaranties that only authenticated and authorized devices may participate in the FL process. This architecture clearly safeguards apps and devices against sophisticated threats.

3.2 Case Study

We give a case study to help a reader better grasp the workflow described above. In a shard-based network, we distribute the blockchain network in shards to avoid a single point of failure (SPoF) and provide scalability to the network. We present this proposed architecture with a case study to help readers better understand real-world scenarios.

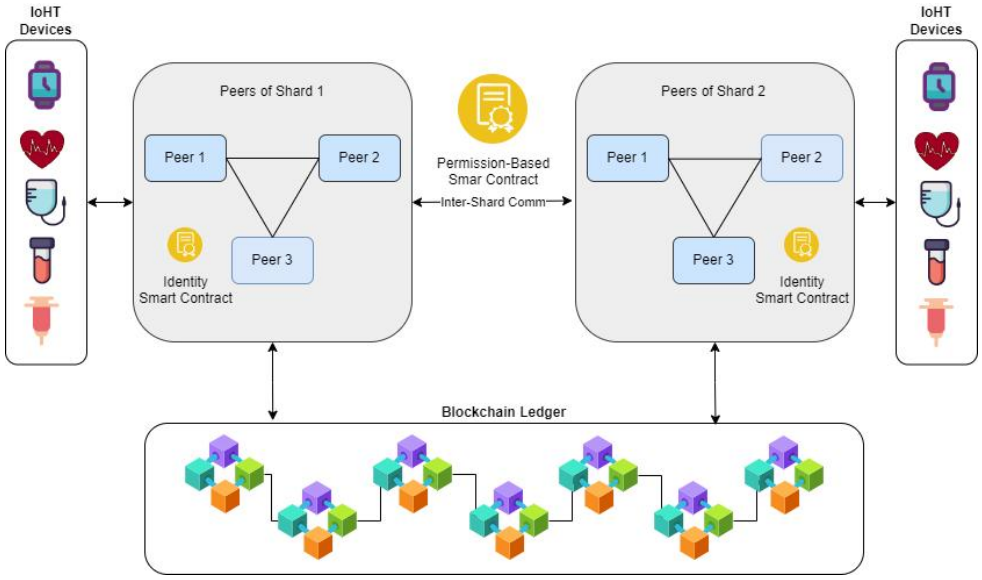

Figure (3.2) Inter Shard Communication and Hyperledger

Shards, peers, and clients/IoHT devices are the three key components that are interconnected in this architecture. In earlier sections, we covered the characteristics and purposes of different components. As a result, it is preferable to map these components to real-world healthcare system scenarios.

Organizations, institutions, hospitals, pharmacies, clinics, and medical laboratories collaborate and interact with each other to transfer, process, and store healthcare data in the healthcare business. As more medical and IoHT devices connect to the network, the security and privacy of these healthcare data are becoming a major problem. As presented in Figure 3.2, this shard-based blockchain network is connected to these healthcare units where different IoHT devices collect healthcare data and participate in the federated learning process.

A single shard can oversee both single and multiple healthcare units. This case study examines a healthcare facility, namely a major hospital, where many departments operate together as peers. The sub-departments participate in the federated learning process by deploying models to client devices and getting updated models in return. As previously said, an individual peer possesses localized blockchain ledgers and off-chain data storage.

A shard can interact with multiple healthcare units, such as small hospitals, clinics, pharmacies, and medical laboratories. A shard can also be used to communicate with the same types of entities; for example, a single block-chain shard can communicate with all pharmacies built in a metropolitan area. In this way, we can create different scenarios to implement this proposed architecture for the scalability of the healthcare network.

3.3 Simulation Setup

The specifications for the simulation testing environment include Ubuntu 20.04 LTS Operating System, Intel(R) Core(TM) i7-10510U CPU at 1.80GHz with a boost up to 2.30 GHz, 512 GB SSD, and 16 GB RAM. Tests are performed in a controlled environment using the specifications described above, to simulate a Fabric test network. The configuration comprises one orderer using the Raft consensus process and eight peers, each affiliated with a distinct healthcare organization, in addition to a certificate authority. We perform these experiments via Hyperledger Fabric, a permissioned distributed ledger system. This platform possesses a modular architecture that facilitates plug-and-play consensus and membership services. Hyperledger Fabric has an execute-order-validate architecture for transactions, allowing each peer to concurrently evaluate models inside each shard. In contrast to other public blockchain platforms that prioritize transactions by their sequence and execute them sequentially.

To simulate the federated learning network, we used the NumPy library without Graphics Processing Unit (GPU) or ML framework. Python version 3.12 is used to simulate the federated learning network. In this network different healthcare devices are simulated that participate in the process and send trained model updates based on the local data.

3.4 Scalability Analysis

Our examination seeks to examine the improvements that are achieved by distributing the transaction load among several shards. Consequently, performance is first evaluated in relation to the number of shards, as seen in Figure 3.3. This workload employs 200 transactions. The system's throughput increases with the addition of more shards. As more shards are added, the network is further divided into smaller networks and each network can manage its own smart contracts and transactions. With more shards, we are creating more opportunities to perform the transactions in parallel rather than sequentially.

The throughput of each independent operating shard is shown in Figure 3.4, which allows a linear increase in global throughput. This directly relates to the time required to assess a model, which is the primary limitation. The Figure 3.5 illustrates that when the submitted Transactions per Second (TPS) increase, the system will eventually reach a saturation level beyond which it cannot support a higher TPS. This may be discerned by analyzing the point in the graph where the average latency begins to increase while the throughput is at its peak. The latency factor decreases with the addition of shards to the layout. Shard 1 has the highest latency and has reached its saturation threshold. The addition of additional shards to the design substantially decreases latency.

It is important to mention that the specified TPS for each number of shards is intentionally set slightly higher than the system's throughput. This is done to fully use the system's capacity throughout the assessment process. Due to the limited computing

Figure (3.3) Shards vs Throughput Analysis

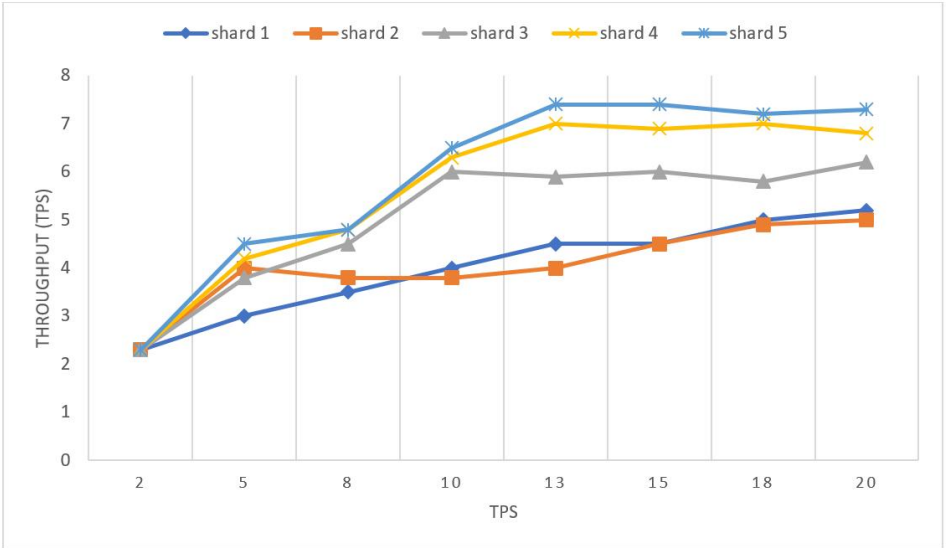

Figure (3.4) TPS vs Throughput

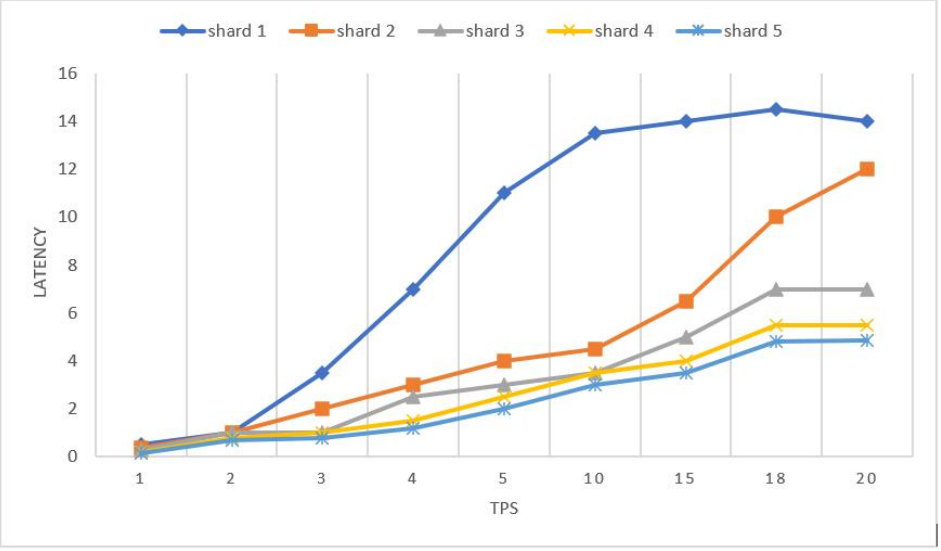

Figure (3.5) Transactions Count vs Latency

resources of the environment, the throughput of each individual shard reaches its maximum capacity and is saturated. By adding a shard one-by-one to the network, the overall throughput of the system increases and goes to a saturation point, whereas increasing the TPS does not increase the overall throughput of the system.

Another view of the throughput vs transaction count is presented in Figure 3.6. Using the mentioned resources and environment, the overall throughput of the system is presented by varying the transactions count. The system's throughput does not grow when it becomes unable to handle the incoming requests due to the overhead involved in processing these extra requests, which restricts the number of successful requests completed. The maximum throughput that can be achieved with these limited resources is 6.8 when the transaction count reaches 400. An increase in the transaction count may decrease the overall throughput of the system and also results in the average increase in latency.

The combination of a sharded blockchain approach with healthcare data security is a promising area for research when a large number of (healthcare) data are processed in a federated learning platform. The proposed scheme is a step forward to the solution of some fundamental issues in federated learning systems based on centralized coordinators. Our scheme gives a way to implement the blockchain-based sharding technique with the FL network to train healthcare data. Where healthcare data privacy and network scalability are major concerns. Individual peer nodes create local ML models based on data stored locally in healthcare equipment. The locally produced models and parameters are shared with other peers via the blockchain platform.

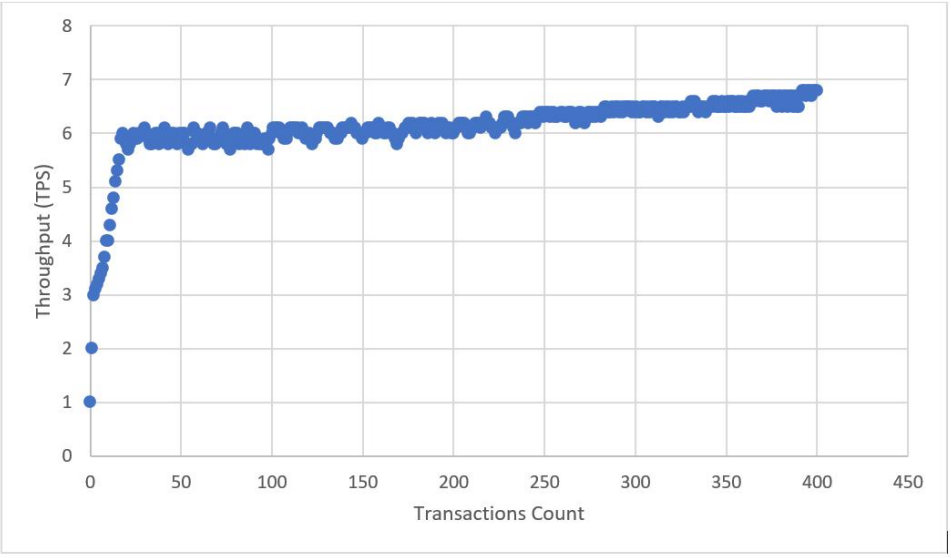

Figure (3.6) Transactions Count vs Throughput

3.5 Conclusion

The integration of a sharded blockchain methodology with healthcare data security presents a viable research avenue for processing substantial volumes of healthcare data inside a federated learning framework. The suggested technique advances the resolution of several basic difficulties in centralized coordinator-based federated learning systems. Our approach provides a method to apply the blockchain-based sharding technology with a federated learning network for training healthcare data. Where healthcare data privacy and network scalability are significant concerns. Individual peer nodes generate localized machine learning models with data saved within healthcare apparatus. The locally generated models and parameters are disseminated to other peers using the blockchain platform.

Chapter 4

Scalability with State Channels

In this chapter, we present a novel layer-2 approach to provide scalability to the federated learning (FL) network. State channel is an off-chain solution to provide scalability to blockchain network. The aim is to minimize on-chain transactions, improve architectural latency, and decrease transaction overhead on the blockchain. These state channels may be used to effectively implement the sharing of federated learning models and to address the scalability issue.

4.1 Introduction

The exchange of healthcare information is increasingly focused on safeguarding data privacy. The rapid progression of communications technology has made the dissemination of fragmented data across several domains an inescapable trend, enhancing data transfer. As noted previously, Federated Learning (FL) is an innovative data-sharing methodology that integrates intelligence with privacy computing and has garnered considerable interest. Although it can facilitate both data value delivery and data privacy protection, it is deficient in protecting against various cyber threats that compromise shared models. Blockchain, a type of distributed ledger technology, is inherently trustworthy; however, it lacks computing capability due to the high latency of its labor-intensive consensus processes.

This chapter presents a strong and reliable architecture, along with a semantic technique for the sharing of FL trained models utilizing state channels, a scaling solution for blockchain technology. The aim is to minimize on-chain transactions, improve architectural latency, and reduce transaction overhead on the blockchain. These state channels may be used to effectively implement the sharing of federated learning models and to address the scalability issue. This study introduces a distributed healthcare federated learning-based safe model-sharing architecture to guaranty healthcare data privacy and scalability.

Health data sharing denotes the transfer of medical and health-related information among individuals, healthcare providers, organizations, and systems, which is crucial for diagnosis, treatment, prediction, and enhancement of medical quality. Various institutions, including healthcare centers, hospitals, smart devices, clinics, and medical research facilities, document and preserve healthcare data. Most healthcare organizations have

transitioned conventional medical records into digital formats [44]. This transformation improves the usability, share ability, and management of healthcare data. The dissemination of health data aids researchers in understand illness patterns, forecast disease trends, and formulate more effective treatment strategies. However, the dissemination of healthcare data can pose several obstacles, encompassing security, privacy, and scalability concerns. Disseminating sensitive information, such as healthcare data, in a trustless environment—particularly when the recipient lacks trustworthiness—entails disregarding several possible dangers, including data leakage, data manipulation, and unauthorized access. Another issue is scalability; as the volume of digitally converted healthcare data increases, the distribution of these data has challenges related to time, bandwidth, and data size. Furthermore, the volume of medical data is anticipated to increase substantially during the next five years. This is attributed to several causes, including the transition of conventional paper healthcare data to digital format, the better use of EHR, advances in healthcare technology, and an increased volume of healthcare data produced by patients and medical equipment [5].

The inherent security properties of blockchain technology, including cryptography, decentralization, and consensus, contribute to the security of blockchain transactions. Blockchain finds utility across various domains, particularly in safeguarding medical health data. This facilitates secure data sharing in an environment devoid of trust, leveraging its features of transparency, traceability, immutability, and invariance. Consequently, the combination of blockchain technology and medical services is currently a prominent subject of interest within research communities [89]. Nonetheless, as the applications of blockchain technology expand, the associated challenges and threats are continually on the rise. A smart contract is essentially a segment of code that is deployed within a public blockchain network, making it accessible to all participants. In the healthcare sector, the dissemination of healthcare information necessitates rigorous adherence to security and privacy regulations [32]. Nonetheless, the implementation of blockchain, especially with Proof-of-Work Consensus (PoW), could pose a performance limitation and requires significant computational and energy resources. As a result, transaction processing times and blockchain throughput are significantly increased (ten transactions per second) compared to custodial payment systems such as Visa, which can manage thousands of transactions per second [95]. Alternative consensus methods, including proof-of-stake Proof of Stake (PoS) and proof-of-federated learning Proof of Federated Learning (PoFL) [64], have been proposed to improve transaction processing speed and minimize latency. Nonetheless, agreement remains a significant challenge that has hindered the widespread adoption of blockchain [1].

In addition, various frameworks have been established to protect privacy during data sharing [99]. FL has attracted considerable attention in the realm of data sharing due to its ability to aggregate data value, protect against data privacy violations, and lessen the burden of network communication. This system inherently preserves privacy by allowing distributed devices within a network to share trained models alongside their local data with a central server or machine while keeping the actual data confidential. This has

been applied in various contexts, such as the Internet of Vehicles Internet of Vehicle (IoV) [39], [59], Industrial Internet of Things (IIoT) [22], and Smart City initiatives [1].

From an analytical standpoint, healthcare organizations primarily use their generated data to train models using machine learning techniques aimed at predicting and detecting disease patterns early. To improve the accuracy of the models, it is essential to gather and combine healthcare data sets from various healthcare domains. This contributes to improving the reliability of the trained model, allowing it to be relevant in various medical conditions, demographics of patients, and healthcare environments.

This chapter explores the integration of permissioned blockchain technology with multiple healthcare organizations, utilizing their localized healthcare data to improve healthcare models using a federated learning approach. Furthermore, we present the idea of state channels to enhance scalability. The increasing collaboration among healthcare organizations in sharing trained models presents significant challenges in registering and validating each transaction, particularly with respect to time, cost, and bandwidth.

Upon sharing a FL-trained model with a requester, the updated model is stored as a transaction on a permissioned blockchain, where the consensus algorithm is employed to validate the transaction. In the context of federated learning, nodes involved in the process engage in iterative sharing of models with a requester. Consequently, validating each transaction of the model update proves to be quite time-consuming for the blockchain. To address this issue, we present the idea of state channels, which serve as an off-chain mechanism to handle each model update trained by the FL network, eliminating the need to interact with the main blockchain for every iteration. Enhances the scalability of the FL network, optimizing the utilization of blockchain technology for improved security, privacy, and scalability.

This key contributions of this chapter are outlined as follows:

- An architecture is developed that integrates blockchain technology with federated learning to enable the secure sharing of trained models. Although FL preserves privacy by keeping raw data localized at participant nodes, it does not inherently guarantee privacy in the model update exchange. Even though raw data is not shared, model gradients can leak private information. Blockchain provides immutable provenance tracking of updates and immutable provenance tracking of updates. Regulations like Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR) prevent healthcare units from sharing sensitive patient healthcare data beyond their own boundaries. FL functions by enabling requester to exchange model parameters instead of healthcare data. The simultaneous achievement of data value transmission and data privacy protection is possible.

- A model storage framework is developed that integrates both on-chain and off-chain components, aimed at enhancing the sharing of healthcare models. Transferring user identification and transaction information to the blockchain enables the achievement

of user identification, authentication, verification of model integrity, and real-time access control to transactions.

- A technique has been developed for executing the FL training process through state channels. This mechanism guarantees that off-chain communication does not lead to any increase in consensus waiting time or impose additional stress on blockchain storage resulting from data sharing and uploading. With the increasing number of healthcare units participating in the FL consortium, the process of communicating model updates may experience delays and inefficiencies. As more networks connect to the system, it improves scalability for model sharing while minimizing interaction with the primary blockchain.

4.2 Related Work

A comprehensive survey [13] that thoroughly examines the developing uses of data security and privacy for FL in the Internet of Healthcare Things (IoHT) networks. They thoroughly examine and assess the primary resolutions to the privacy and data security concerns IoHT. A scheme [77] examines the suitability of a distributed reinforcement learning approach in an FL network and evaluates the possible advantages of integrating blockchain technology in the distributed system. The proposed approach improves clinical monitoring and ensures secure communication and data privacy in a decentralized manner. A survey [48] examines the difficulties, solutions and future paths to effectively use blockchain in FL contexts with limited resources.

In a study [52], the authors explain the significant impact of blockchain-based federated learning on improving efficient healthcare. They highlight the crucial function of blockchain-managed edge nodes to prevent any potential failure with the usage of FL. An extensive survey work specifically on the state channels has been presented in [55]. The authors discussed the integration of state channels with the blockchain to improve the scalability of the network.

A survey work has been presented to improve the scalability of the blockchain [66]. The authors discussed the payment channel and transaction rollup techniques to improve the scalability of blockchain transactions in the financial sector. A technique has been presented to improve the computational and supervision capabilities of the blockchain using state channels [98]. Using state channels to generate sandboxes and instantiate federated learning tasks to implement a trust supervision mechanism based on these sandboxes.

A technique known as StateFL has been introduced, a modified Blockchain Federated Learning (BCFL) architecture that employs state channels to alleviate the burden on the blockchain by decreasing the volume of on-chain transactions, improving the latency of the system and reducing transaction costs accordingly [31].

4.3 State Channel Model Sharing Architecture

This section of the chapter examines a prevalent situation in which several healthcare providers cooperate to exchange trained models derived from healthcare data. Each entity possesses its own healthcare data resources at its particular location. To resolve this issue, we have developed a model-sharing system utilizing Federated Learning that incorporates a Permissioned blockchain with state channels. In this architecture, a healthcare entity may refer to a hospital, research center, medical laboratory, clinic, or intelligent healthcare unit. There are three principal role factors for the architecture.

4.3.1 FL Model Requester

A FL model requester denotes entities that need the training of a FL model utilizing healthcare data from other healthcare entities, contingent upon mutual agreement of terms and conditions, as illustrated in the Figure 4.1. A research center or hospital might provide this duty under our system. The FL model requester initiates the model sharing request to the blockchain. The FL model requester might function as a representative system for the aforementioned healthcare entity, overseeing all model sharing requests. The system must authenticate its identity via the blockchain to submit a model-sharing request.

4.3.2 The Blockchain Network

The blockchain network is maintained and managed by all intermediaries. According to the situation described in this article, we utilize a Permissioned blockchain to construct the architecture for exchanging models. We deployed the Hyper Ledger Fabric (HLF) for this purpose.

4.3.3 Federated Learning Resource Network

FL participant nodes are required to be registered on the Permissioned blockchain, irrespective of the device type. The registration information comprises the identity, location, and kind of recorded data. This information will be recorded as transactions on the HLF upon verification. A header node in the resource network can initiate model sharing activities, aggregate or quantize models, and train models for other connected nodes. The header nodes provide distinct purposes for various activities. A model-sharing position often begins with a single model requester and encompasses many healthcare organizations. FL develops a centralized model utilizing healthcare data from several institutions. We established the FL resource network configuration in Kubernetes, enabling the deployment of nodes within containers for efficient management and scalability.

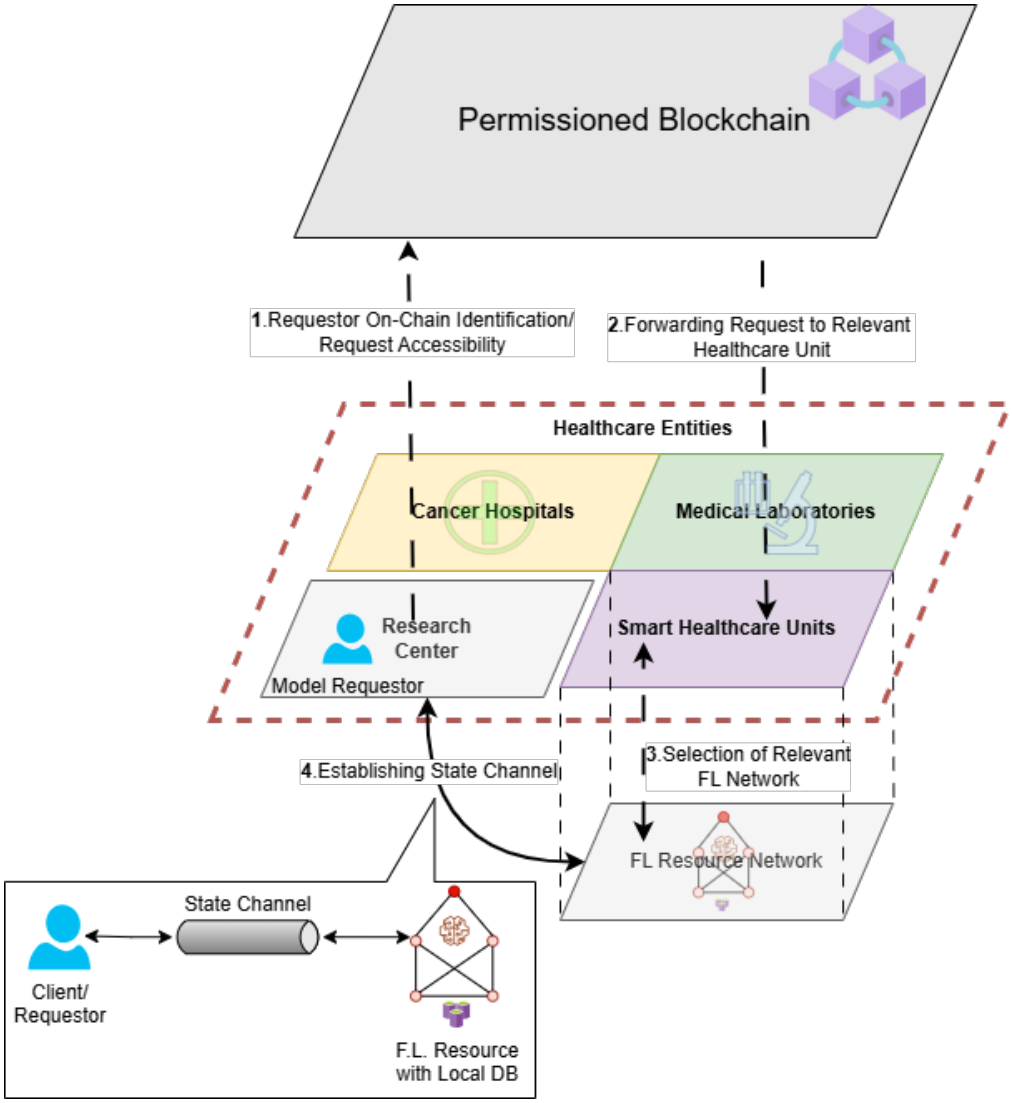

Figure (4.1) Blockchain-based FL trained model sharing architecture. Different healthcare entities share FL-based healthcare data trained model over the state channels.

4.3.4 Data Storage Structure

This approach has two data storage structures: on-chain storage and off-chain storage. In federated learning, data is retained in local storage, and only the trained models derived from the local datasets are transmitted to a central entity. A second significant characteristic of FL is the decentralized data architecture that mitigates SPoF Single Point of Failure and protecting user privacy.

There are two types of data stored in this structure:

1. Local Database: Healthcare data is stored in healthcare devices managed and maintained by healthcare entities, and only trained models are exchanged. To simulate the healthcare devices network to participate in the FL process, we utilized the Kubernetes architecture, where the FL participating nodes are initialized in containers. It becomes easy to scale, automatic deployment of nodes, and management of these containerized nodes.

2. Blockchain Database: Important information, such as latest model update, metadata of trained models like training results, cryptographic hashes of trained models, etc. is stored on the HLF, to realize data integrity verification and traceability of transactions.

4.4 Scalability of Networks

The state channel is an advanced variant of blockchain technology that improves efficiency and scalability by executing some transactions off-chain. Creating a state channel across healthcare institutions for the transmission and computation of federated learning-based models on local devices decreases transaction confirmation time, lowers transaction costs, and reduces reliance on on-chain storage.

The training process consists of three steps:

1. State-Channel Creation

2. Off-Chain Training Iteration

3. On-Chain Settlement

4.4.1 Creation of State Channel

Upon consensus between the model requester and the FL resource network, the FL resource nodes obtain the training parameters to commence the FL training process. A state channel is created between the FL resource header node and the model requester node. A correlation between the secured on-chain state and the off-chain transactions is established, ensuring that activities within the state channel do not influence the blockchain state. Initially, they may look for existing channels; but, if none are accessible, they will start a new channel.

Technical Details for Creating State Channel

The objective for two healthcare organizations (e.g., Cancer Hospital and Research Center) is to securely exchange trained models and evaluate them without paying on-chain gas fees or transaction delay. Upon verifying the request to train the model via the blockchain, all parties concur on the terms and conditions; a smart contract or chaincode is executed on the blockchain, endorsed by both parties, encompassing all terms and conditions for the initiation and termination of a state channel. We have provided a general description of state channels in the previous chapter. Here, we discuss some key features and differences of state channels with the IPC (inter-process channel) and RPC (Remote Procedure Call) channel. State channels are off-chain communication channels like IPC or RPC but they are different in trust, cryptography, and dispute resolution. IPC channels allow processes to communicate within the same machine, such as pipes, sockets, and message queues. RPC channels also enable communications with remote machines if they are in a local network. However, state channels are considered as secure channels where participants exchange signed updates and the blockchain acts as a final settlement or dispute resolution layer. State channels provide properties that RPC or IPC cannot provide, such as authenticity, dispute resolution, and finality. State channels provide off-chain computation with verifiable state transition. We utilized the Python gRemote Procedure Calls (gRPC) framework to construct state channels, functioning within a client-server paradigm. We have installed the 'grpcio' and 'grpcio-tools' libraries. In this suggested architecture, the model requester entity can operate the server to obtain model updates from the FL resource network. The server side manages the state channel logic, including channel administration, state updates, and dispute resolution. The client-side entity engages with the state channel server for model dissemination and validation. A protocol file with the '.proto' extension is utilized by both the client and server sides to establish a state channel between two entities. We installed Python version 3.11.9 to facilitate the client-server approach.

Submitting Chain Code to Permissioned Blockchain

This design employs the HLF framework as a permissioned blockchain, with smart contracts, referred to as chaincode in HLF, installed within it. A chain code is published on the blockchain to establish a state channel for two healthcare institutions to exchange model updates derived from federated learning. This chain code initializes and oversees the state channel, retains information, and anchors the federated learning process for accountability and dispute resolution. This chain code governs the life cycle of the state channel and retains information like the channel ID, the starting model state or version, the specifications for the trained model (quality, epochs, and iterations), and security primitives.

4.4.2 Off-Chain Training Iteration

In this setup, during each cycle, local nodes obtain the initial global model from the header node and submit updated models trained on local datasets. To finalize one iteration, the header node executes the aggregation procedure and transmits the aggregated model to the requester node via the established state channel. The number of iterations may be established at channel setup, and the channel may be terminated if the training results do not meet the requester's requirements. All interactions between two healthcare entities, such as model updates, feedback, and retraining, transpire off-chain via the established state channel. Each new state provides the revised model hash H(M1), quality, timestamp, and is digitally signed by both parties.

Subsequent to each cycle, a header node transmits a global model to the designated node, therefore finalizing an off-chain transaction. Minor transactions, such as model exchanges in each iteration, are recorded off-chain, alleviating the burden on the blockchain network. The most recent model is sent to the blockchain in the last iteration. Healthcare organizations may exchange models with FL-Networks off-chain without interfacing with the primary blockchain. This diminishes the necessity for frequent, high-volume interactions with the primary blockchain, hence enhancing scalability. gRPC employs protocol buffers for data serialization, guaranteeing a concise and efficient procedure. Furthermore, it operates on the Hyper Text Transfer Protocol (HTTP)/2 protocol, which facilitates multiplexing, compression, and bidirectional streaming during transmission.

4.4.3 On-Chain Settlement

Upon completion of the FL procedure, a party wishing to terminate the channel must send the most recent model update to the HLF for On-Chain settlement, duly signed by both sides. Should one party choose to terminate the channel following several iterations due to discrepancies or unauthorized model upgrades, that party may cease operations and petition the HLF for settlement by submitting the most recent valid trained model state.

There may be two justifications for On-Chain settlement and the termination of the channel: Concluding the channel subsequent to the fulfillment of the FL mission. Termination of communication in the event of a conflict. The initial reason is that on-chain settlement commences upon the validation of the cryptographic hash of the trained model's final state. Both participants calculate the hash of the final aggregated model and confirm the hash value. Upon confirmation, the HLF acquires this hash value and begins the transaction validation procedure. If any participants identify faults or unauthorized modifications, the most recent signed model update is uploaded to the blockchain to initiate dispute resolution. The implemented chain code authenticates the provided state using signatures and timestamps. On the HLF, a valid state is recorded as a last state.

Figure 4.2 illustrates the establishment of a state channel between two healthcare entities utilizing the gRPC architecture for the iterative exchange of trained models inside

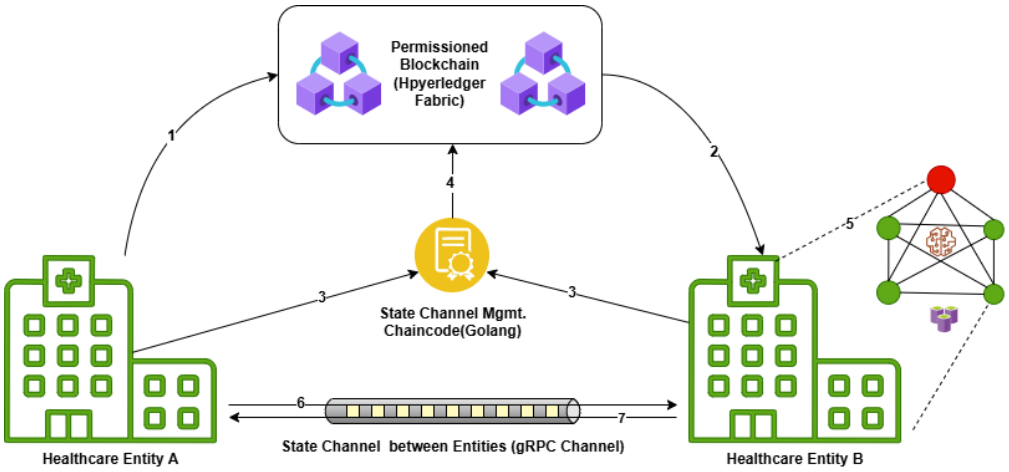

Figure (4.2) Establishing of state channel between two healthcare entities to share trained models

the FL resource network. We outline the procedures necessary to create a state channel between two healthcare organizations.

1. Healthcare entity A submits a request to the HLF to train a model by providing information such as the type of model architecture, the number of parameters, and the entity-B ID, along with its own identifying details.

2. The HLF conducts an identification verification of entity-A and evaluates the request in accordance with established procedures. Verifies if this request including shared data is allowed for forwarding to entity B in accordance with established regulations. If permitted, transmit the request to entity-B along with the request specifics.

3. Both parties establish terms and conditions (e.g., number of epochs, model quality, model type, data type) within the chain code, which is digitally signed by both parties.

4. Upon configuring the chain code, it is deployed on the Hyperledger Fabric and a state channel is formed between entity A and entity B.

5. Entity B initiates the training process via its federated learning resource network.

6. In each iteration, entity B disseminates an aggregated model over the state channel (off-chain) accompanied by its digital signature, without engaging with the HLF.

7. Upon successful receipt of the modified model in each iteration, entity A transmits an acknowledgment to entity B, accompanied by its digital signature.

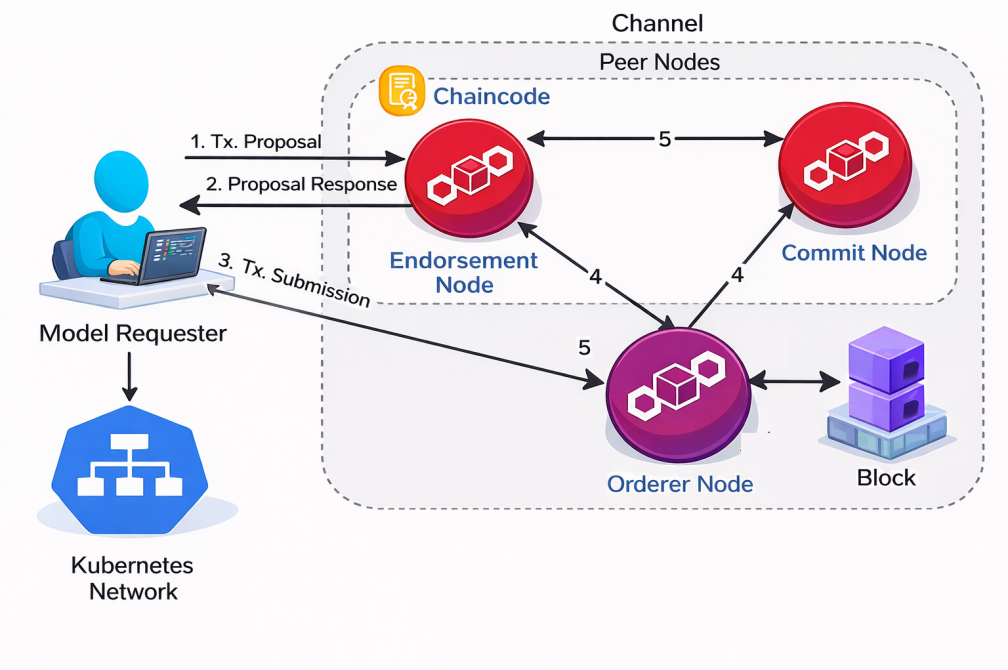

Figure (4.3) Peers Communication with Raft Consensus Algorithm

4.5 Evaluation and Results

4.5.1 Scalability Analysis

To assess scalability, we examine two parameters: bandwidth and the duration required to finalize activities using state channels and blockchain. Figure 4.3 presents two graphs, one illustrating the overall time required to accomplish activities in relation to a fluctuating number of entities. An entity denotes healthcare institutions, including hospitals, pharmacies, medical research facilities, etc. Each entity comprises 50 nodes and does tasks over 10 iterations. Incorporating blockchain into task management renders the process more time-intensive. This occurs as each entity immediately logs its model modification into the primary blockchain, leading to an escalation in transaction volume. As a result, the consensus method requires more time to validate these transactions. This results in prolonged reaction times owing to the development of blocks. Utilizing state channels allows organizations to communicate with the primary blockchain intermittently, hence considerably decreasing latency while transactions are validated off-chain. The second plot decreases bandwidth use by employing state channels, wherein changes are exclusively sent between entities having finalized version upgrades on the blockchain. In the primary blockchain, bandwidth consumption escalates owing to the continual transaction recording for each model change of every entity.

4.5.2 Communication Architecture

This section succinctly examines the communication architecture of the HLF among peers involved in the transaction validation process. This discourse is pertinent since it addresses

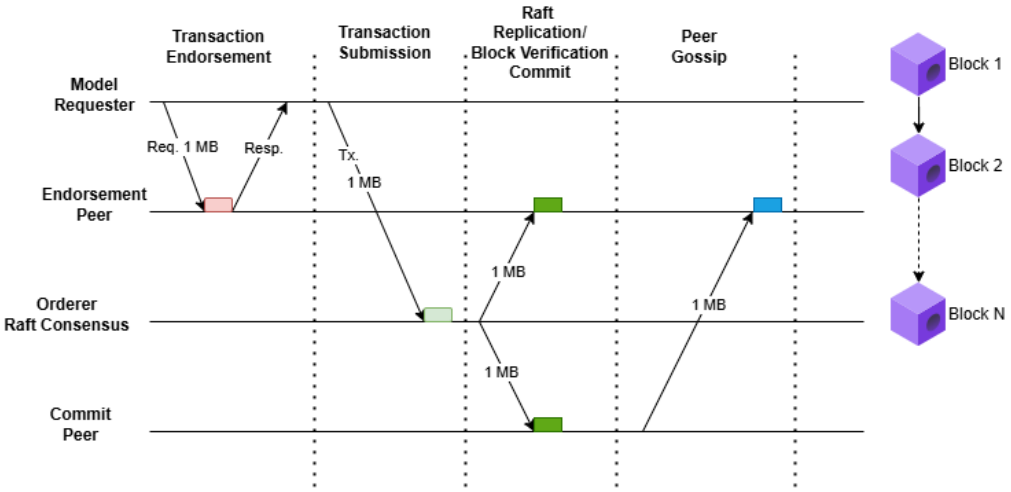

Figure (4.4) Consumption of bandwidth in Raft consensus algorithm.

the scalability challenge of blockchain technology, characterized by the extended consensus time necessary for transaction validation. In the event of several iterative transactions, such as changes to the FL models, both time and bandwidth rise substantially.

Figure 4.3 illustrates three nodes in the Hyperledger Fabric (HLF) that participate in the model validation process after the completion of the federated learning (FL) task: 1. Endorsement node, 2. Orderer node and 3. Commit node. Upon completion of the job, the model requester node transmits the transaction proposal (1 in Figure 4.3) to the endorsement node. The state channel management chaincode is further implemented at the endorsement node. The endorsement node transforms this proposal into a transaction after the validation of the proposal from the deployed chain code and returns it (2 in Figure 4.3) to the model requester node. The model requester node subsequently transmits this transaction to the orderer node (3 in Figure 4.3) together with the modified model, resulting in the creation of a block. Subsequently, the orderer node transmits the generated block to the endorsement and commit nodes, who then record the block in the ledger. Subsequently, the block is finalized in the HLF.

The endorsement peer nodes accept the proposal and generate transactions. Nevertheless, the commit peer nodes or non-endorsement nodes do not engage in transaction origination and only partake in transaction validation upon block commitment. Assume that a trained model of 1 MB is presented to the HLF for elucidation. The model requester node transmits the 1 MB trained model to the endorsement peer node (Transaction Endorsement in Figure4.4) to initiate a transaction execution and validation process. The execution of chain code converts the supplied proposal into a transaction after validation. The supporting peer node cryptographically signs the transaction using its private key. The model requester obtains the transaction from the endorsing peer and transmits the 1 MB trained model included within the transaction to the orderer node, where the Raft consensus process is operational (Transaction Submission in Figure.4.4). Generate a block and duplicate it to the endorsement and commit nodes (Raft Replication in Figure 4.4). The supplied model is duplicated twice for validation at the endorsement and commit

nodes. The HLF comprises one endorsement, one commit, and one orderer node. The final step (Peer Gossip in Figure 4.4) disseminates the generated block to all peer nodes until every peer has received it. Thus, validating a trained model with a size of 1 MB will require 5 to 6 MB of bandwidth on the HLF for validation.

4.5.3 Heterogeneous Model Architecture

In the scenario under discussion, multiple healthcare entities are involved, each sharing their different trained models from various healthcare devices. The reason is the different computational and memory resources of the healthcare devices. Some devices, such as wearable, have very limited computational resources, and some healthcare devices, such as thermal cycler machine or Magnetic Resonance Imaging (MRI) machines, may have GPU's built into their architecture. We are utilizing four distinct model architectures to train the models for healthcare devices. These are One-Dimensional Convolutional Neural Network) (1DCNN), Gated Recurrent Unit (GRU), MobilNetV2, and ResNet-18. In Table 4.1, we present a comparison of these model architectures with their generated and quantized model sizes for each iteration of the FL sharing process.

Table (4.1) Different Model Architecture with Different Model Sizes

Model Arch.	**No. of Param.**	**Model Size (MB)**	**Quantized Size**
IDCNN	500K	1 -2	0.5 - 0.8
GRU	1M	2 - 5	1 - 2
MobilNetV2	3.4M	10 - 14	3.5 - 5.6
ResNet-18	11M	30 - 40	8 - 11

4.5.4 Model Sizes and Scalability

The trained model's size is reduced in each iteration using quantization. This study examines the potential model size for each iteration of the federated learning process in several models and evaluates how state channels might enhance scalability regarding time and bandwidth relative to blockchain technology [56]. The HLF often employs the Raft consensus method by default for validation reasons. It is lightweight and exhibits optimal performance when several nodes engage in the consensus process. This discussion focuses on the duration and bandwidth utilized for sharing a federated learning model developed across the state channel and the blockchain. When using the state channel, the conclusive version of the trained model is authenticated on the blockchain, but not for each iteration. If the network speed is 100 Mbps (Megabits per second) or 12.5 MBps (Megabytes per second), the upload time for the GRU model architecture, with a model size of 1 to 2 for each iteration, may be computed.

In federated learning, the size of the updated model is contingent upon the number of parameters and the precision of these values, commonly represented as 32-bit floats,

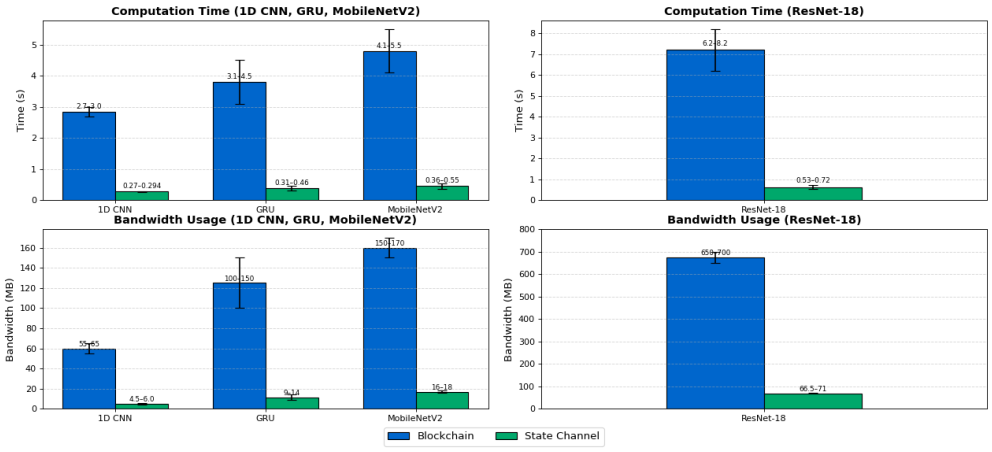

Figure (4.5) Blockchain and State Channel Time and Bandwidth Comparison for different Model Architecture.

equating to 4 bytes per parameter. Certain architectures employ 16-bit floating-point representations for accuracy, which dramatically reduces model sizes. For instance, the 1DCNN model design often involves the exchange of 500,000 parameters to train the revised model. If each parameter is 4 bytes, utilizing 32-bit float precision, the model size is about 500,000 x 4 bytes, equating to 2,000,000 bytes. Approximately 2MB.

The quantity of devices in FL is irrelevant, as each device independently trains its local model and subsequently distributes the updated parameters. The aggregator node gathers information from all devices and constructs a global model of uniform size and structure in each iteration. This evaluation compares the utilization of blockchain with state channels to blockchain without state channels. Consequently, the actual time expenditure to validate a shared model occurs during the execution of a blockchain transaction validation method. We are comparing the suggested design with the default HLF architecture based on two parameters: time and bandwidth. Figure 4.5 illustrates the assessment of computation time and bandwidth utilization between the HLF and state channel methods across several model architectures: 1D CNN, GRU, MobileNetV2, and ResNet-18, demonstrating a distinct performance superiority of state channels in federated learning contexts.

Figures 4.6 and 4.7 illustrate the efficacy of the state channel when we incorporate more entities, each utilizing 30 to 50 healthcare devices and training models with varying topologies. It is evident that as the number of healthcare firms increases, the HLF requires more time and bandwidth to validate transactions compared to state channels.

4.6 Conclusion

This chapter presents a scalability solution through the integration of state channels with the Hyperledger Fabric blockchain. Diverse healthcare organizations trained models on their varied devices via federated learning. The trained models are shared repeatedly

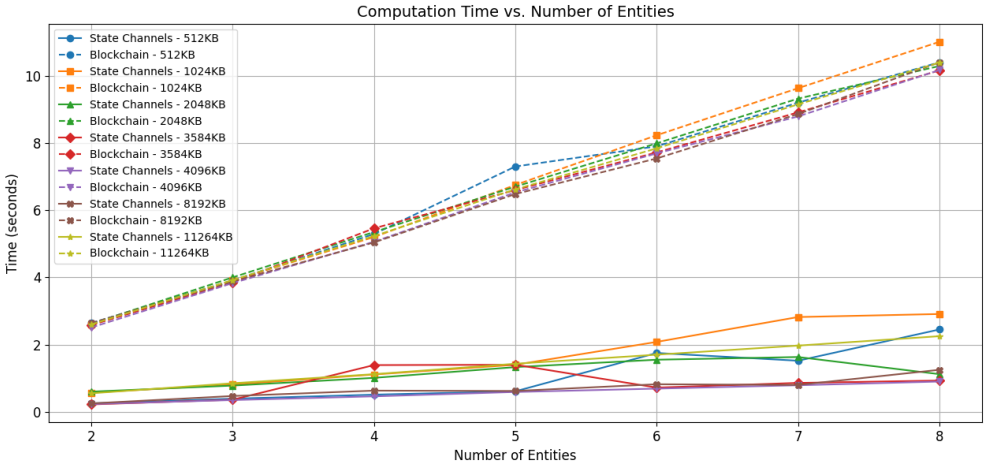

Figure (4.6) Blockchain and State Channel Time Comparison with Increasing Healthcare Entities.

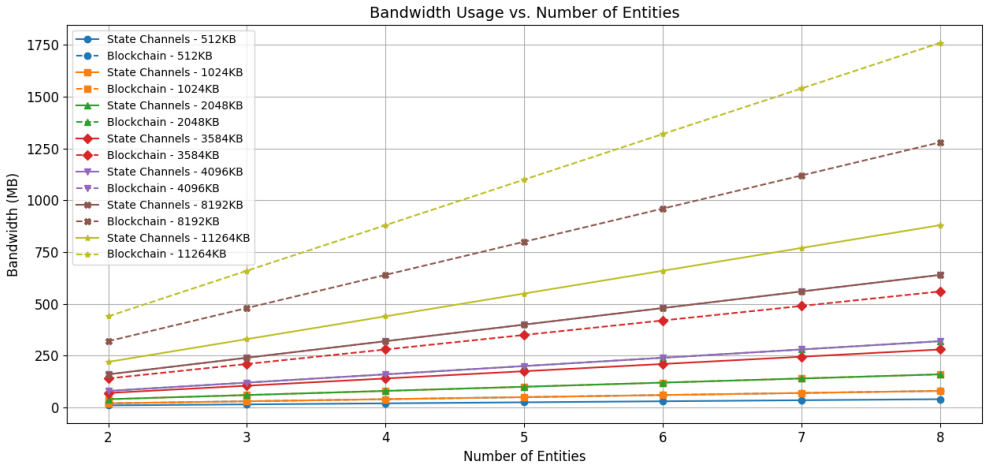

Figure (4.7) Blockchain and State Channel Bandwidth Comparison with Increasing Healthcare Entities.

using existing state channels to enhance scalability and reduce blockchain consensus time and use of network bandwidth. The results ultimately reveal considerable benefits regarding time and bandwidth when employing state channels with blockchains.

Chapter 5

ABAC model integration with Blockchain with Trust Attribute for Security of Federated Learning Networks

This chapter introduces an innovative and reliable access control mechanism for the dissemination of healthcare-oriented FL models using blockchain technology. This segment of the research introduces a distributed federated learning-based model sharing architecture designed to safeguard the privacy and security of healthcare data through the implementation of an Attribute-based Access Control (ABAC) model, where the trust attribute is computed and used for access control decisions. In addition, it offers a trust monitoring mechanism for the dissemination of remote healthcare models to identify rogue nodes within the federated learning network.

5.1 Introduction

Every individual in our digitally sophisticated world demands adequate data protection. The transmission of healthcare information is increasingly based on the protection of data privacy. The fast progression of communications technology has made the dissemination of fragmented data across several domains an inevitable trend, enhancing data transport. FL is an innovative data-sharing methodology that integrates intelligence with privacy computing, garnering considerable interest. Although it can deliver both data value and data privacy, it lacks oversight throughout the application process, undermining the reliability of the calculation process and the transmission of results. Blockchain, being a type of distributed ledger technology, is inherently trustworthy.

This chapter introduces an innovative and reliable access control mechanism for the dissemination of healthcare-oriented FL models using blockchain technology. This segment of the research introduces a distributed federated learning-based model sharing architecture designed to safeguard the privacy and security of healthcare data through

the implementation of an ABAC model, where the trust attribute is computed and used for access control decisions. In addition, it offers a trust monitoring mechanism for the dissemination of remote healthcare models to identify rogue nodes within the federated learning network.

Health data sharing denotes the transfer of medical and health-related information among individuals, healthcare providers, organizations, and systems, which is crucial for the diagnosis, treatment, prediction, and enhancement of medical quality. Various institutions, including healthcare centers, hospitals, smart devices, clinics, and medical research facilities, document and preserve healthcare data. Most healthcare organizations have transitioned conventional medical records to digital formats [45]. This change facilitates the sharing and management of healthcare data. The dissemination of health data aids researchers in comprehending illness trends, making forecasts, and formulating enhanced treatment strategies. However, the dissemination of healthcare data may pose several obstacles, including security, privacy, and access control concerns. Distributing sensitive information, such as healthcare data, in a trustless environment, particularly when the receiver lacks trustworthiness, entails disregarding several possible dangers, including data leakage, data manipulation, and unauthorized access [6]. In addition, the volume of medical data is expected to rise substantially during the next five years. This phenomenon is ascribed to multiple factors, including the transition from traditional paper-based healthcare data to digital formats, the heightened utilization of (EHR), advances in healthcare technology, and the increasing volume of healthcare data produced by patients and medical devices [67].

In recent generations, intelligent healthcare systems have provided advanced services. Remote patient monitoring, enabled by the IoHT, enhances healthcare services for people in rural and disadvantaged regions. IoHT devices assist healthcare professionals to supervise medical equipment, managing the hospital environment, and automating routine assessments. These features may improve operational efficiency and reduce costs [43]. However, these services are also raising security, privacy and access control concerns as an increasing number of IoHT devices produce and disseminate data.

Cyber criminals have discovered alternative strategies to exploit digital healthcare data, that go beyond merely selling it in bulk to illicit marketplaces. Considering this, further precautions are required to safeguard healthcare data [23]. In 2023, the Office of Civil Rights (OCR) officially recorded a total of 725 instances of data breaches. These breaches led to the exposure or unauthorized disclosure of over 133 million records [33]. Another study indicates that over 100 million persons were impacted by cyberattacks on healthcare organizations in 2023. In 2022, the assaults affected approximately 44 million people [65]. The financial ramifications of these breaches are substantial. In 2022, the average cost of a healthcare data breach increased to 10.10 million dollars, up from 9.23 million dollars in 2021. Healthcare data breaches are the most costly in all sectors [69].

5.1.1 Blockchain Technology for Healthcare

Blockchain technology possesses intrinsic security features, including cryptography, decentralization, and consensus, which ensure the security of blockchain transactions. Blockchain is applicable in several domains, particularly in the realm of medical heath data security and privacy protection. Facilitates safe data exchange in a trustless context through its qualities of transparency, traceability, immutability, and invariance. The integration of blockchain technology with healthcare services is now a prominent subject of investigation among research communities [90]. However, as the uses of blockchain technology expand, its problems and risks are always increasing. A smart contract is a code segment put on a public blockchain network, accessible to everybody. In the healthcare industry, the dissemination of healthcare information requires strict adherence to security and privacy rules. Consequently, safeguarding privacy in healthcare data sharing requires robust access control along with the integration of blockchain inside the healthcare industry.

From a research point of view, primarily healthcare organizations utilize their generated data to train models that employ machine learning techniques to predict and identify early trends of illness. To develop more precise models, it is essential to collect and amalgamate healthcare datasets from many healthcare domains. Enhances the resilience of the trained model, making it useful in various medical conditions, demographics of patients, and healthcare environments.

5.1.2 Security of Healthcare Data

To ensure dependable and safe dissemination of the healthcare model across several domains, it is essential to tackle the security concern of trust monitoring in multiple aspects, including data storage, user identification, and training outcomes. We want robust transparency and immutability features that blockchain technology can offer [50]. To ensure detailed access control on users' requests for model sharing, we employed the ABAC paradigm developed by National Institute of Standards and Technology (NIST) [28]. The trust management module is incorporated to furnish trust attributes to the ABAC model, therefore excluding malevolent nodes from participating in FL tasks.

This chapter discusses the application of permissioned blockchain technology via the HLF framework inside a consortium of several healthcare organizations, utilizing their local healthcare data to train healthcare models through a federated learning approach. To achieve more precise access control between the model requester and the federated learning resource network, we deployed the ABAC model on the blockchain to augment the security of the architecture.

The objective of this chapter is to provide a trust supervision mechanism for FL that aims to control the behavior of nodes (IoHT devices) and use this trust value as an attribute for the ABAC model to control the flow of information.

- We develop an architecture that combines blockchain technology with federated learning to facilitate the secure sharing of trained models. Due to regulations such as

HIPAA and GDPR, healthcare units cannot share sensitive patient healthcare data outside their units. FL operates by having requester exchange model parameters rather than the healthcare data. Data value transmission and data privacy protection can be achieved simultaneously.

- We develop a model storage framework that encompasses both on-chain and off-chain components, with the goal of facilitating healthcare model sharing. By transferring user identification and transaction information to the blockchain, it is possible to achieve user identification, authentication, verification of model integrity, and real-time access control to transactions.

- We design two smart contracts/chaincodes based on the HLF platform, namely, the Identity Management Module smart contract and Policy Decision Module (PDM) smart contract.

- To detect and isolate the malicious node in the FL resource pool network, we propose a trust management system integrated with the ABAC model that calculates trust values based on the quality of the trained model, the behavior and participation frequency of the node. All healthcare units must trust that all participants are contributing legitimate data models and following agreed-upon protocols. Based on the calculated trust value for a node, the management decides to select a node to participate in the FL data sharing task and isolate the nodes that are acting maliciously.

5.2 Related Work

Federated Learning (FL) has been extensively utilized to provide safe data transfer while maintaining privacy. Simultaneously, the examination of trust monitoring in FL has attracted considerable focus. Diverse potential solutions have been suggested within the relevant fields.

The growing proliferation of IoT devices is making their management increasingly difficult in terms of scalability and access control. An attribute-based access control architecture utilizing the Hyperledger Fabric blockchain is suggested to address these difficulties [75]. The ABAC concept eliminates the need for creating access control lists Access Control List (ACL) or assigning roles to every system user. ABAC provides access contingent upon the traits exhibited by the subjects. The suggested architecture employed the Hyperledger Fabric Raft consensus mechanism for transaction verification due to its superior speed and simplicity in achieving consensus compared to the Kafka ordering service. A conventional consensus Proof of Work (PoW) method has been enhanced with the smart contract-based ABAC model to provide decentralized, granular, and adaptive access control management in Internet of Things (IoT) contexts [100]. The enhanced PoW technique markedly reduces energy usage and transaction validation time.

Using blockchain technology to address security issues in IoT networks introduces

hurdles such as significant overhead, elevated energy consumption, and substantial costs for transaction validation. A trust-based access control system, referred to as TABI [58], is introduced for the edge IoT network. It uses access and trust assessment techniques connected to the Hyperledger Fabric blockchain to reduce the influence of malevolent IoT users and devices.

Sharing health information without the patient's consent is constrained by regulatory requirements for healthcare data. Inter-Planetary File System (IPFS) facilitates the management of healthcare data, while the ABAC model is incorporated with permissioned blocks to ensure access control over patient information [68]. This solution facilitates emergency access to patient data while adhering to regulatory restrictions. A centralized ABAC system presents issues of a single point of failure and restricted transparency, attributable to reliance on a single server, which may result in data vulnerabilities. Blockchain can address this issue through its decentralized architecture and offer precise access control for IoT devices. The trust value functions as an additional criterion for evaluating the user reputation and is reassessed after each contact to allocate differing access capabilities, thus enabling dynamic rights management [92].

5.2.1 Blockchain and Federated Learning

A detailed review [14] that meticulously explores the evolving applications of data security and privacy in Federated Learning inside the Internet of Healthcare Things (IoHT) networks. The authors start with a thorough introduction to the essential ideas of Federated Learning (FL) as applied within the framework of the Internet of Health Things (IoHT). They meticulously evaluated and analyzed the principal solutions to the privacy and data security issues of the IoHT. In addition, they categorize the most significant papers related to the security of IoHT data. Subsequently, many IoHT network datasets are enumerated for the objective of model training. They underscore the essential conclusions from this assessment, concentrating on the current challenges and prospective directions for future research in protecting data security and privacy in IoHT networks using Federated Learning (FL).

The authors [78] assess the suitability of a distributed reinforcement learning methodology within a multidisciplinary reinforcement system in federated learning and analyze the potential benefits of incorporating blockchain technology into the distributed framework. Blockchain-enabled reinforcement federated learning facilitates the management of post-COVID-19 patient data from Internet of Health Things applications without reliance on intermediary dependencies and transactions. The suggested method improves clinical oversight and ensures safe communication and data confidentiality in a decentralized fashion. The main objective is to improve the efficiency and scalability of the federated learning reinforcement process in a distributed environment, while safeguarding data confidentiality and security with blockchain technology for the Internet of Healthcare Things (IoHT) applications.

A survey [49] investigates the challenges, solutions, and prospective avenues for the

effective implementation of blockchain in federated learning environments with limited resources. They provide a comprehensive analysis of several blockchain technologies applicable to the FL process and evaluate their trade-offs concerning a limited resource budget. Furthermore, they meticulously analyze the cyber threats that may arise in a resource-constrained federated learning environment and elucidate the crucial function that blockchain might fulfill in mitigating such incursions.

Research [53] discusses the substantial influence of blockchain-based federated learning (FL) in enhancing healthcare efficiency. They underscore the essential role of blockchain-managed edge nodes in averting possible failures, while also accentuating the use of federated learning by Internet of Health Things devices to efficiently leverage dispersed clinical data. They evaluate the benefits and limitations of merging both systems via a content analysis approach. The authors emphasis three principal domains of inquiry that have been thoroughly examined concerning blockchain technology: 1) Internet of Healthcare Things (IoHT); 2) management of electronic health records (EHRs) and electronic medical records (EMRs); and 3) secure notifications within digital healthcare systems (internal consortium). In addition, they present a novel conceptual framework for employing blockchain technology to facilitate federated learning within the digital healthcare domain.

The openness feature of the blockchain poses privacy concerns [101]. A suggested design mitigates this risk by storing private data and limiting access to authorized people exclusively. Upon validation of IoT device transactions by blockchain-endorsing peers, the participant's cloud archives the data in accordance with the ABAC paradigm.

5.2.2 Blockchain and Healthcare

A sophisticated healthcare system introduced in [79] that emphasizes energy conservation and privacy using a blockchain-enabled federated learning framework. Nevertheless, WBAN users may hesitate to share data without enough incentives, while miners may express concerns over the significant energy expenditure required to sustain the blockchain. An optimization problem is established with the aim of maximizing the system's overall efficiency, including aspects such as energy consumption, incentives for the Wireless Body Area Network (WBAN), miner revenue, and losses in FL. This work employs Differential Privacy (DP) and Homomorphic Encryption (HE) approaches to protect against information exposure. This is accomplished by adding noise to gradients before changing model weights and encrypting the outcomes prior to transmitting them to miners.

A comparable storage architecture is utilized [57] to enhance data dependability and fortify authorship protection. The integration of IPFS with blockchain creates a traditional storage framework that encompasses both on-chain and off-chain elements. This framework is widely used as a reliable instrument for data storage. A number of experts have analyzed the efficacy of traceability, energy consumption, and several other concerns about its application.

The health data sharing methodology using hybrid blockchain (HSHB) is presented in [88], which categorizes sharing transactions into data sharing on the private chain and the alliance chain based on distinct sharing entities. HSHB has developed a health data access control strategy to prevent conflict of interest between various institutions, based on their identification and access requirements. HSHB offers customers an effective and stable query mechanism for historical health data by constructing a B+ tree index of the health data.

A comprehensive medical information sharing system utilizing blockchain technology is suggested [11]. This system gathers IoT-based patient health information from various non-invasive medical devices. A confidential medical data exchange system utilizing cloud servers and a proxy re-encryption method to enhance the confidentiality of medical information. They utilized a Hyperledger Fabric architecture for a permissioned blockchain to facilitate data management and access control. The assessment of the proposed system relies on the computational overhead test, performance evaluation, and security analysis.

5.3 Blockchain with Federated Learning

This chapter examines a prevalent situation in which several healthcare organizations cooperate to exchange trained models derived from healthcare data. Each entity possesses its own healthcare data resources at its particular location. To resolve this issue, we have developed a model-sharing system utilizing Federated Learning that incorporates a Permissioned blockchain. In this architecture, a healthcare entity may refer to a hospital, research facility, medical laboratory, clinic, or advanced healthcare unit. These entities disseminate healthcare data for various objectives. Three principal role factors exist for the architecture.

- **FL Model Requester** An FL model requester refers to entities that need to train an FL model based on healthcare data from any other healthcare entity, as shown in Figure **??**, a research center or hospital could play this role in our framework. The FL model requester starts with the model sharing request to the blockchain. The FL model requester could serve as a representative system for the healthcare entity discussed previously, managing all model sharing requests. This system should verify its identity from the blockchain to issue a model-sharing request. It also negotiates the incentives with the FL resource network and releases the incentives to the FL resource network after completion of the tasks.

- **The Blockchain Network** The blockchain network is maintained and managed by all intermediaries. The blockchain is a decentralized transaction ledger that securely maintains all verified transactions. Typically, these transactions are kept in blocks using a Merkel tree structure. The blocks next to each other are connected at their ends to create a chain. The blockchain refers to the longest sequence of interconnected blocks. The immutability and distributed nature of this tool make it highly effective in enhancing the performance of current solutions for situations such

as crowd-sourcing [103]. According to the management authority, the blockchain can be categorized into three types: public blockchain, private or Permissioned blockchain, and alliance blockchain. The last two are characterized by a smaller scale and a stricter management mechanism compared to public blockchain.

- **Federated Learning Resource Network** A data client node has a single interactive channel to the model-sharing system, known as the "proxy." The proxy device might be either the client's personal smart device or an edge server. Regardless of the device's kind, it must be registered on the permissioned blockchain in order to function as a proxy. The registration information primarily consists of the user's identification details, address details, and model resource details. Once verified, this information will be documented on the permissioned blockchain as transactions. A proxy can act on behalf of clients to carry out various operations in the model sharing process, including starting model sharing tasks, executing model training. The function of a single proxy may vary between distinct jobs. A model-sharing task often begins with a single model requester entity and involves the participation of numerous healthcare entities. The objective of FL is to train a centralized model using data from several healthcare entities. This model is then used to fulfill model sharing requests and provide predictions on relevant issues.

5.3.1 Work Flow

This part will provide a detailed explanation of the model-sharing procedure utilizing the permissioned blockchain and ABAC, alongside model training through the FL methodology. Figure 5.1 illustrates several healthcare entities, including research centers, hospitals, medical labs, and advanced healthcare units. Each business produces and manages many forms of healthcare data pertinent to their activities. Nonetheless, they are capable of executing machine learning procedures on their produced datasets. However, if a trained model is required from another institution, the model requester may solicit it from the other healthcare organization. Nonetheless, other security dangers are linked to model transfer, including model poisoning attacks, inference attacks, model inversion attacks, and Byzantine failure attacks [96]. In this framework, a model requester submits a request to the permissioned blockchain to train a model by specifying its needs and parameters. The blockchain authenticates the requester and transmits the request to the appropriate healthcare agency. The pertinent healthcare practitioner chooses the appropriate FL network to execute FL duties. We assume that each model provider inside the architecture possesses their own local datasets. They utilize their local health data to train the models and engage in model-sharing activities. Figure 5.2 illustrates the workflow of the model sharing procedure.

Step 1: A model requester node initiates a model sharing request to the permissioned blockchain with model training parameters. This request describes the requirements for training the model like: rounds per epoch, batch size, learning rate, source diversity, etc.

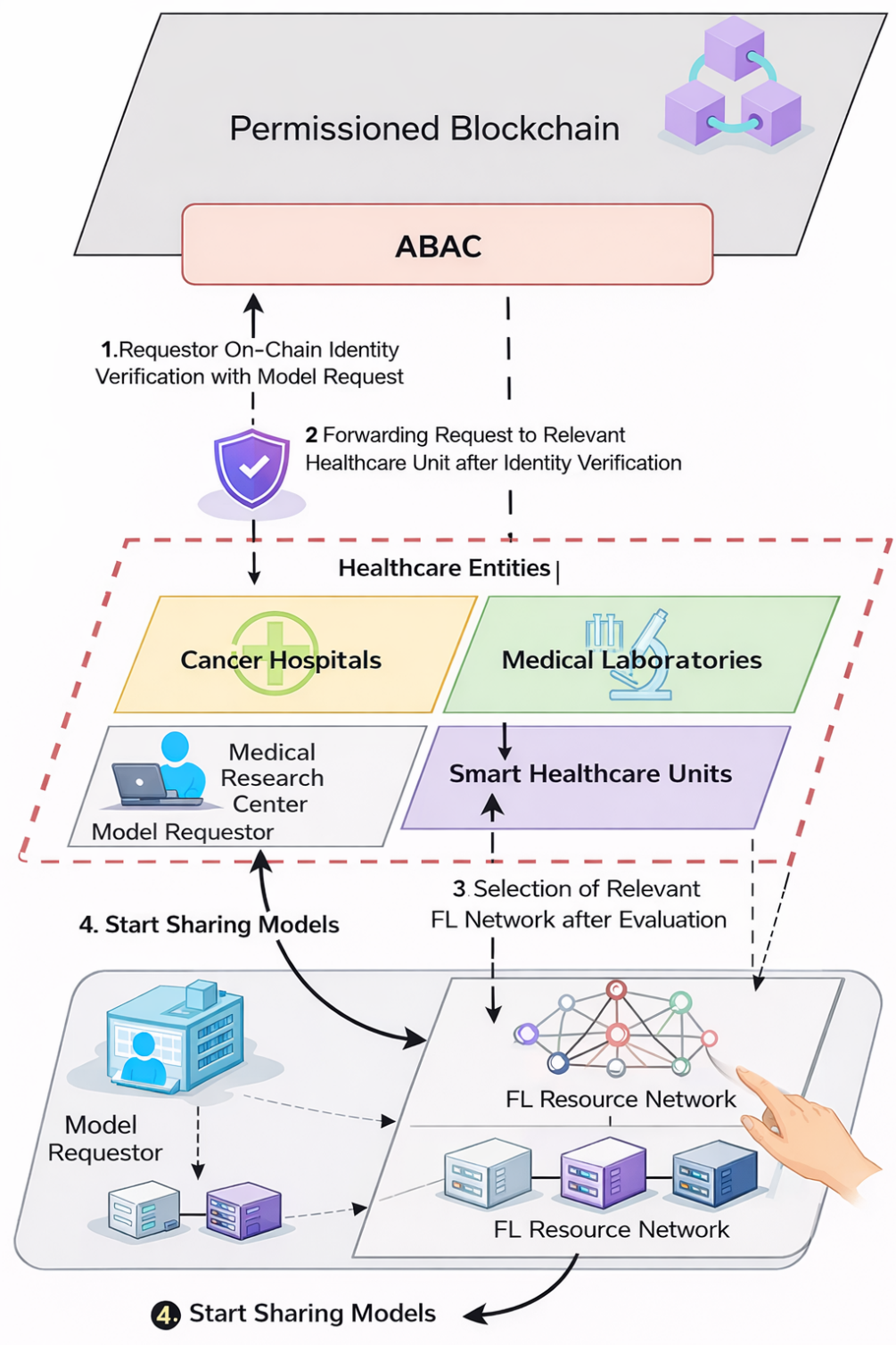

Figure (5.1) ABAC model integration with Blockchain for FL Access Control.

Step 2: The permissioned blockchain receives the request to start FL process and verifies its authenticity. The verification process encompasses not only the validation of the request's format, but also authenticates the requester identity and the legality of the request.

Step 3: The permissioned blockchain examines the request and selects the appropriate FL network to create a relevant group of nodes based on the parameters and the proxy registration information stored on the blockchain.

Step 4: Blockchain forwards the required information to train the model with all parameters to the selected FL network.

Step 5: This is a crucial step; the request is evaluated with the provided attributes and checked against the access control policies. If the request passes the policy check, then the nodes are filtered based on the trust values of the selected network.

Step 6: The selected FL network performs the model training process based on the parameters received with their local databases.

Step 7: The FL resource network sends the trained results by encrypting them on each iteration over the established communication channel. For each iteration, the requester node analyzes the received results, checks the model quality according to the requirements, and releases an incentive/reputation value for the successful task. Within the established communication channel, both communicating parties can perform a number of operations, such as updating the trained model without involving the blockchain.

Step 8: After completing the locked tasks, the FL resource network sends an update about the task completion to the blockchain, as well as to the model requester.

Step 9: The blockchain distributes the incentives according to the performance values of the nodes to use them for other tasks and registers a transaction in its register about task completion.

5.3.2 Incentive Distribution

Offering incentives to participating nodes is essential to encourage them (FL nodes) to contribute their models and computing resources for the training process. The incentives may encompass several forms: financial rewards, resource credits, access to models, reputation enhancement, performance-based awards, etc. In this architecture, a model requester offers reputation-based incentives to the participating nodes following the completion of federated learning tasks. The Shapley Value (SV) has been extensively utilized to evaluate the quality of the FL node model [97]. In FL, an efficient strategy has been presented [81] to attract nodes with a high-quality model by utilizing SV to assess the contributions of various nodes in FL. The SV, derived from cooperative game theory, quantitatively assesses the contribution of each member to the global model. Nodes with elevated SVs exert a more substantial impact on model performance and, hence, should be allocated a correspondingly larger share of the incentives. An elevated reputation can result in enhanced trust, precedence in forthcoming assignments, and more exposure.

To calculate the SV for a participating node in FL, we need to measure the contribution

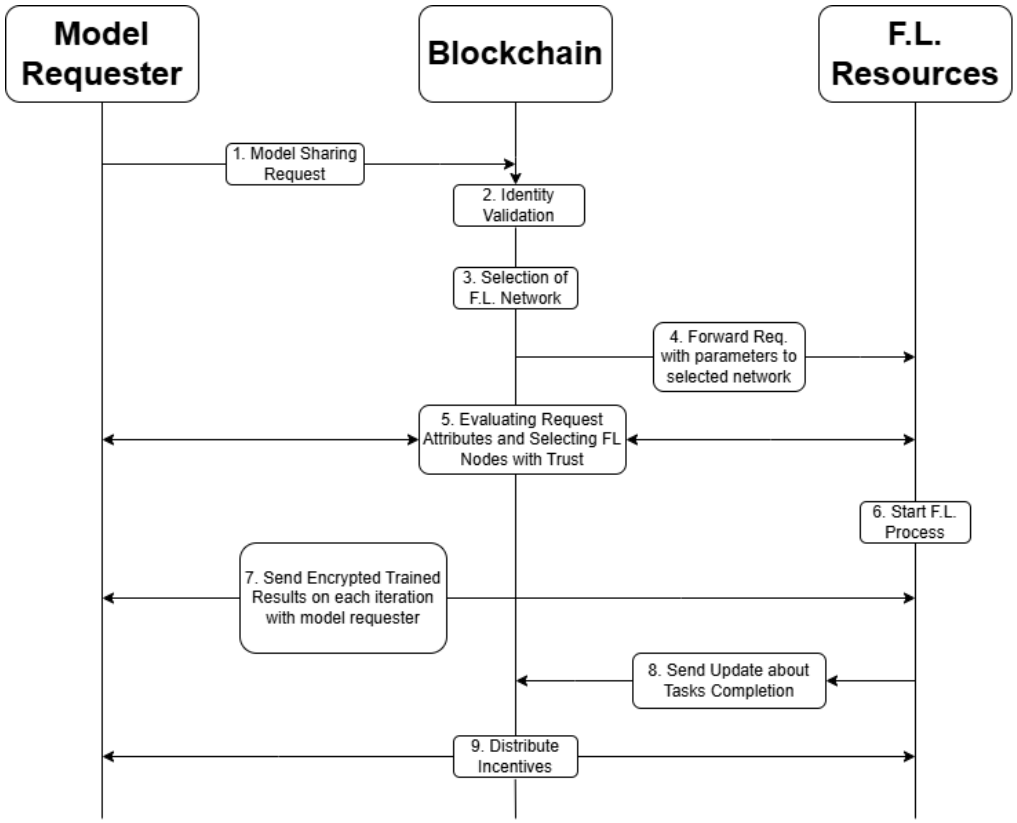

Figure (5.2) Request Validation and Model Sharing Steps

of each node to the overall performance of the model. It also determines the value of each node's data for training the global model. To calculate the SV, we consider the M number of nodes in a FL resource pool. The accuracy function $\nu(S)$ of the global model presents the performance of a subset of(S) nodes used for training, where $(S) \subseteq M$. For all subsets $(S) \subseteq M$, we need to compute $\nu(S)$, which can be a performance factor of the global model trained on the data provided by the nodes in (S). In each iteration, we train the global model in each possible subset (S) of nodes and determine the accuracy function $\nu(S)$ of the trained model for all possible subsets $(S) \subseteq M$. For example, if there are 3 nodes, the models will be trained with subsets: $\{\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$.

For each node N_i its accuracy in a subset (S) of nodes where $(i \notin S)$ can be determined by the difference between the accuracy of the trained model with and without the node as described in the following equation:

$$Accuracy(N_i) = \nu(S \cup i) - \nu(S)$$

It describes how the contribution of node N_i to the subset (S) affects the precision of the model.

SV is calculated for node N_i by averaging its contribution across all possible subsets (S) of nodes excluding i.

$$\Phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{S!(M-S-1)!}{M!} \times [v(S \cup \{i\}) - v(S)] \tag{5.1}$$

S is a subset of nodes excluding N_i

$\nu(S)$ is the accuracy of the global trained model with nodes in S.

$\nu(S \cup \{i\}) - \nu(S)$ is the marginal contribution of N_i

$\frac{S!(n-S-1)!}{n!}$ ensures fairness across all possible node orderings.

Calculate SV Φ_i for each node i, showing their contribution to the performance of the global model. To convert the SV's into reputation values, we can use the following formula:

$$R_i = R_b + k\frac{\Phi_i}{\sum \Phi} \tag{5.2}$$

R_i is the new reputation value for node i. k is the scaling factor to control SV in the reputation. Determines the importance of the SV in the overall reputation. A larger k increases the impact of the node's contribution on the reputation value. In other words, it adjusts the weight or influence of the contribution component (normalized SV) in the calculation of the reputation value R_i. In this case, the value of $k = 1.0$. R_b is the base reputation value with which all nodes start. In this case $R_b = 1.0$. $\sum \phi$ is the sum of all SV's of all nodes. normalizing the contribution.

To understand the reputation-based incentive distribution with the help of SV, we simplified it by taking an example of three nodes that participate in the federated learning process and training models based on their local data. The SV formula is used to fairly distribute the incentives among participants.

Contribution of Nodes with an Example

In this section, we explain the working of the SV algorithm with the help of an example. Suppose that the initial model accuracy was 70% before sharing it with the participating nodes. After completion of the federating learning training process, the three participating nodes achieve accuracy: $N_1 = 78\%$, $N_2 = 81\%$, and $N_3 = 85\%$. Now, to understand the SV formula for the three participating nodes, the accuracy of the model can be determined by the combination of nodes. After aggregating the trained models with FedAvg formula, the overall accuracy of the aggregated model increases due to the collaboration of nodes, complementary data distribution, and non-linear impact on the accuracy. However, for simplicity, we are not taking these factors into account to get a working understanding of the SV formula and incentive distribution. Marginal contributions of nodes 1, 2 and 3 are described in Tables 5.1, 5.2 and 5.3 respectively.

Formula to calculate SV for three nodes

$$\phi_i = \frac{1}{3!} \sum_{all\ subsets} Marginal\ Contribution$$

Node 1 Shapley Value:

Table (5.1) Marginal Contribution of Node 1.

Subset S	$\nu(S)$	$\nu(S \cup \{1\}) - \nu(S)$
$\{1\}$	$70\% \to 78\%$	8%
$\{2\} \to \{1,2\}$	$81\% \to 79.5\%$	-1.5%
$\{3\} \to \{1,3\}$	$85\% \to 81.5\%$	-3.5%
$\{2,3\} \to \{1,2,3\}$	$83\% \to 81.33\%$	-1.67%

Table (5.2) Marginal Contribution of Node 2.

Subset S	$\nu(S)$	$\nu(S \cup \{2\}) - \nu(S)$
$\{2\}$	$70\% \to 81\%$	11%
$\{1\} \to \{1,2\}$	$78\% \to 79.5\%$	1.5%
$\{3\} \to \{2,3\}$	$85\% \to 83\%$	-2%
$\{1,3\} \to \{1,2,3\}$	$81.5\% \to 81.33\%$	0.17%

$$\phi_1 = \frac{1}{3} \cdot 8 + \frac{1}{6} \cdot (-1.5) + \frac{1}{6} \cdot (-3.5) + \frac{1}{3} \cdot (-1.67) = 1.28$$

Node 2 Shapley Value:

$$\phi_1 = \frac{1}{3} \cdot 11 + \frac{1}{6} \cdot (1.5) + \frac{1}{6} \cdot (-2) + \frac{1}{3} \cdot (-0.17) = 3.53$$

Node 3 Shapley Value:

$$\phi_1 = \frac{1}{3} \cdot 15 + \frac{1}{6} \cdot (3.5) + \frac{1}{6} \cdot (2) + \frac{1}{3} \cdot (1.83) = 6.52$$

Formula to calculate SV for three nodes

$$\phi_i = \frac{1}{3!} \sum_{all\ subsets} \textit{Marginal Contribution}$$

Node 1 Shapley Value:

$$\phi_1 = \frac{1}{3} \cdot 8 + \frac{1}{6} \cdot (-1.5) + \frac{1}{6} \cdot (-3.5) + \frac{1}{3} \cdot (-1.67) = 1.28$$

Node 2 Shapley Value:

$$\phi_1 = \frac{1}{3} \cdot 11 + \frac{1}{6} \cdot (1.5) + \frac{1}{6} \cdot (-2) + \frac{1}{3} \cdot (-0.17) = 3.53$$

Table (5.3) Marginal Contribution of Node 3.

Subset S	$\nu(S)$	$\nu(S \cup \{3\}) - \nu(S)$
$\{3\}$	$70\% \to 85\%$	15%
$\{1\} \to \{1,3\}$	$78\% \to 81.5\%$	3.5%
$\{2\} \to \{2,3\}$	$81\% \to 83\%$	2%
$\{1,2\} \to \{1,2,3\}$	$79.5\% \to 81.33\%$	1.83%

Node 3 Shapley Value:

$$\phi_1 = \frac{1}{3} \cdot 15 + \frac{1}{6} \cdot (3.5) + \frac{1}{6} \cdot (2) + \frac{1}{3} \cdot (1.83) = 6.52$$

Reputation Points (RP) of Nodes

After calculating the SV's of the nodes, the formula to calculate reputation-based incentives is given in Equation 5.2.

The total sum of SV's of three nodes:

$$\sum \phi = 1.28 + 3.53 + 6.52 = 11.33$$

Now, reputation values for three nodes:
Node 1 incentive = $\frac{1.28}{11.33} = 1 + 0.113 = 1.113$ RP
Node 2 incentive = $\frac{3.53}{11.33} = 1 + 0.312 = 1.312$ RP
Node 3 incentive = $\frac{6.52}{11.33} = 1 + 0.575 = 1.575$ RP

5.4 Access Control for Federated Learning

5.4.1 Importance of Integrating ABAC with Blockchain

Integrating an ABAC model with blockchain is not just a technological advantage but also a strategic improvement that augments security, compliance, and scalability. The ABAC approach offers dynamic control at the attribute level, contingent upon several aspects including user attributes, resource attributes, and contextual attributes, hence facilitating fine-grained access control.

In a consortium blockchain, participant management may necessitate regular updates to role configurations across all peers using classic RBAC; each new member may require such adjustments. ABAC demonstrates superior scalability, since once a policy administrator establishes policies and rules, a new participant with relevant characteristics instantly acquires the appropriate access, hence reducing administrative burden and facilitating streamlined policy administration.

5.4.2 Data Storage Structure

To share learned models, secure storage and efficient data management, together with integrity verification, are essential. Blockchain can address these challenges through its cryptographic capabilities. This approach has two data storage structures: on-chain storage and off-chain storage. In federated learning, data is retained in local storage, and only the trained models derived from the local databases are transmitted to a central entity. A second significant characteristic of federated learning is the decentralized data architecture that mitigates the risk of a single point of failure and safeguards user privacy. The unprocessed data are not transmitted across the network, hence mitigating the hazards of data exposure and augmenting user privacy.

There are two types of data stored in this structure:

1. Local Database: Healthcare data is stored in the local database managed and maintained by healthcare entities, and only they can control data access permissions to their database.

2. Blockchain Database: Important information, such as data abstract, training results, cryptographic hashes of trained models, etc. is stored on permissioned blockchain, to realize data integrity verification and traceability of transactions.

5.4.3 Data Access Control Architecture

Upon comprehensive evaluation of the model-sharing context and the necessity for a dependable service, it is imperative to establish an ABAC model to enhance and enforce tight access control for the sharing of trained models. To achieve this objective, we can refer to the ABAC definition and considerations established by NIST [28].

This ABAC model has eight tuples (MR, R, D, M, T, N, S, I) whose elements, respectively, represent the Model Requester, Resource, Data, Model, Task, Node, Security, and Incentive. The meaning of these attributes is defined below.

1. Model Requester Attributes: Typically, it contains details such as the identity of the model requester requesting access.

2. Resource Attributes: In general, it includes information about the requested resource.

3. Data Attributes: Data on different nodes may vary in nature and reflect various healthcare information such as Electrocardiogram (ECG) signals, images, and text-based data. Furthermore, the sensitivity of the data may range between nodes, depending on factors such as data size and quality.

4. Model Attributes: This information encompasses several strategies employed to aggregate model updates from disparate nodes. Hence, these aggregation models must include the ability to regulate malevolent or defective updates.

5. Task Attributes: These attributes are specific to the work at hand. We can classify the machine learning task into three types: supervised learning, unsupervised learning, or reinforcement learning.

6. Node Trust Attributes: The properties of the nodes are associated with their trustworthiness and behavior, such as their model quality, learning rate, delay factor, and reputation values.

7. Security Attributes: The type of encryption method used for the confidentiality of the model and the identity verification of the participating nodes determines the security attributes.

8. Incentive attributes: These attributes pertain to incentive processes, such as the allocation of reputation values or the ranking of participating nodes.

We utilize Smart Contracts (SCs) to provide critical functions, such as request management, data storage, and decision-making for the access control system. We are employing Hyperledger Fabric to implement the permissioned blockchain. In the Hyperledger Fabric platform, smart contracts are referred to as chaincodes. Consequently, we employ the word chaincode to denote smart contracts. The chaincodes are developed in Go language version 1.20. The Hyperledger Fabric documentation provides a comprehensive tutorial on packaging, installing, committing, and invoking chaincode [18]. These chaincodes are implemented on the channels established between two communicating entities. Two primary smart contracts, or chaincodes, are integral to the identity management module and access control functions. These are the Identity Management Module (IMM) chaincode and the PDM chaincode. The chaincodes are implemented on the primary blockchain and utilize characteristics as input to enforce the policies of the ABAC architecture. The functionality of the IMM chaincode is delineated in Algorithm 1 (CheckAccess) and Algorithm 2 (Grant Access). The ABAC model is constructed by merging the IMM and Trust Management Module (TMM) with the NIST-defined ABAC model, as seen in Figure 5.3. This ABAC model has two categories of components: on-chain and off-chain. The on-chain elements of the ABAC operate on the blockchain, whereas the off-chain components do not retain policy and attribute data on the blockchain.

The functions of each module in the model are as follows:

1. **Identity Management Module** This module verifies the identities of the registered model requester and authorizes authenticated participants to perform activities. Access control policies are based on the attributes of the Model Requester (MR), Resource (R), Data (D), and Security (S).

2. **Policy Enforcement Module** The task of this module is to transform the access request received from the model requester, which includes its attributes, and forward it to the PDM to obtain the result of the decision for the request. Upon receiving the result, the policy enforcement module will grant or deny the access request based on the instructions provided by the policy decision module.

3. **Policy Decision Module** The module is responsible for determining authorization outcomes by evaluating access control policies and data attributes and determining whether to grant or deny requests. In this framework, this module works on-chain and a chaincode is deployed to take the decisions according to the set policies. An algorithm of PDM chain code is presented below with the title Algorithm 4 (Policy Decision Module Chaincode.

4. **Policy Administration Module** The module is responsible for the storage and management of access control policies, as well as providing a basis for the decision-making process in the policy decision module.

5. **Policy Information Module** The module is responsible for storing and managing attribute information. It interacts with the user trust management module (TMM)

Algorithm 1: IMM-Chaincode

```
INPUT:Model Requester ID, Resource ID, accessType
OUTPUT:Access Allowed or Not

function RegisterUser(UserID, Role)
    Create User Object (UserID, Role)
    Store User on Blockchain
    Return "User Registered Successfully
end function

function GrantAccess(PolicyID, RequesterRole, RequestedData, ModelType,
SecurityType, RequesterID, ResourceID)
    Create Object (PolicyID, RequesterRole, RequestedData, ModelType,
SecurityType, RequesterID, ResourceID)
    Store Policy on Blockchain
    Return "Access Policy Created"
end function

function CheckAccess(UserID, RequestedData, ModelType, SecurityType,
ResourceID)
    Retrieve User Details from Blockchain
if User does not exist then
|    return
     "Access Denied";
    Retrieve all policies from Blockchain
    FOR EACH policy in policies
if Policy.RequesterRole == User.Role AND
Policy.RequestedDataType == RequestedDataType AND
Policy.ModelType == ModelType AND
Policy.SecurityType == SecurityType AND
Policy.SubjectID == SubjectID AND
Policy.ObjectID == ObjectID then
     return "Access Granted";

    END FOR
    Return "Access Denied"
end function
```

to get information about the user trust attribute. This information is used by the policy decision module (PDM) to make a policy decision and the attribute-based access control module to create access control requests. Attributes such as Task(T), Security (S), and Model (M) are stored in a repository integrated with this module. Collects the attributes of the node (N), trust (T) and data (D) attributes from the TMM and uses these attributes to take decisions according to the applied policy.

6. **Trust Management Module** It is responsible for managing the trust attribute information of the participating nodes stored on the permissioned blockchain and providing it to the PIM to isolate bad or malicious nodes in the FL resource network. We will discuss the functions of the trust management module in the upcoming section. In this proposed ABAC model, the TMM is our point of interest that calculates the trust values of the nodes working in the real-time network and forwards these values as an attribute to the PIM.

Algorithm 2: Policy Decision Module Chaincode

INPUT: `policyID, modelRequester, resource, dataType, modelType, task, nodeTrustValue, security, incentive`
OUTPUT: `Decisions: Yes on No`

function AddPolicy(policyID, modelRequester, resource, dataType, modelType, task, node, security, incentive)
4: **Create** `ABACPolicy with given attributes`
Convert `ABACPolicy to JSON format`
Store `JSON in blockchain with key = policyID`
Return `"Policy Added Successfully"`
8: **end function**

function CheckAccess(modelRequester, resource, dataType, modelType, task, nodeTrustValue, security, incentive)
Retrieve `all stored policies from blockchain`
FOR `each policy in blockchain:`
12: **if** *policy matches request attributes:* **then**
| **return**
`"Access Granted";`
End FOR
Return `"Access Denied"`
end function

16: **function** GetPolicy(policyID)
Retrieve `policy from blockchain using policyID`
if *policy exists:* **then**
Convert`JSON to readable format`
20: `policy details`
Else: return `"Policy not found";`
end function

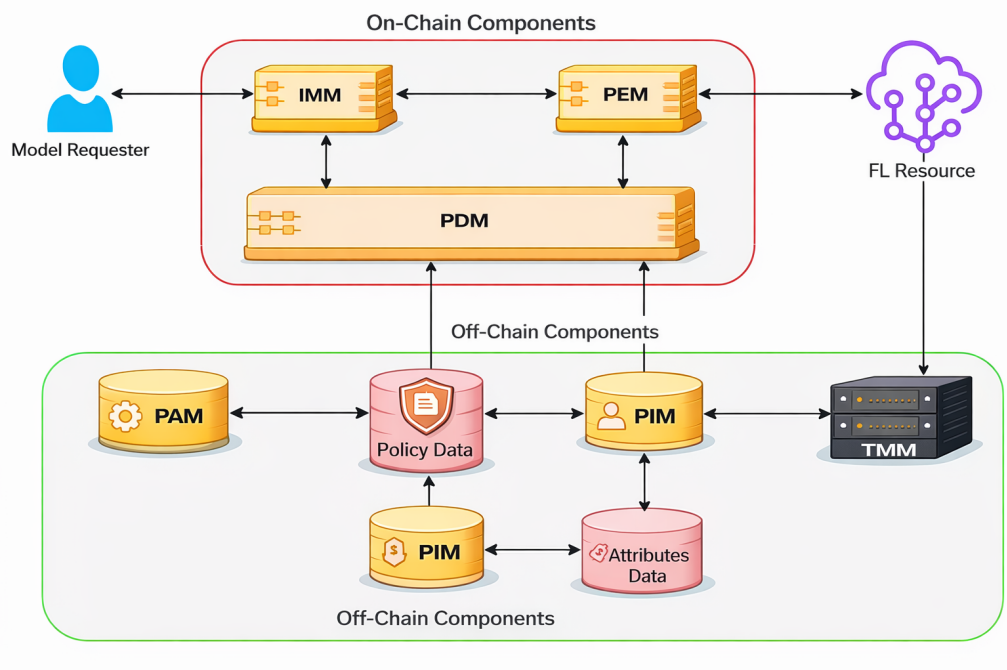

Figure (5.3) Data Flow Model for ABAC

Figure 5.4 presents an alternative perspective on the administration and execution of chain codes. This graphic aids in comprehending both off-chain and on-chain procedures. PAM functions as a policy administrator and oversees access control policies. It implements the IMM and PDM chain codes to execute identification and access policies. The Resource network supplies all pertinent data necessary for calculating the trust levels of the FL nodes in relation to the PIM. The access control method utilizes PAM, PIM, and TMM modules, which operate off-chain to store policy, attributes, and trust data, respectively.

5.4.4 Chaincodes for IMM and PDM

In this ABAC model, we create the chaincodes for the IMM and PDM modules. The purpose of creating chaincodes for these two modules and implementing them on the permissioned blockchain is to demonstrate the automation process of the ABAC model and to minimize needless overhead on the blockchain. Additional modules such as PAM and PIM operate off-chain. The PAM module serves as the administrative component and necessitates human intervention by an administrator to oversee the policies. The PIM module is utilized for the storage and management of attribute information and does not necessitate extensive automation. Consequently, the IMM and PDM modules are the primary components that must be implemented on the blockchain to facilitate effective access control. The algorithms for these chaincodes are delineated in Algorithms 1 and 2. The combination of ABAC with blockchain facilitates effortless policy enforcement across both off-chain and on-chain systems. A healthcare organization can utilize current ABAC rules to regulate blockchain transactions. Enhances compatibility with current systems. Furthermore, it offers robust transparency and auditability for the access

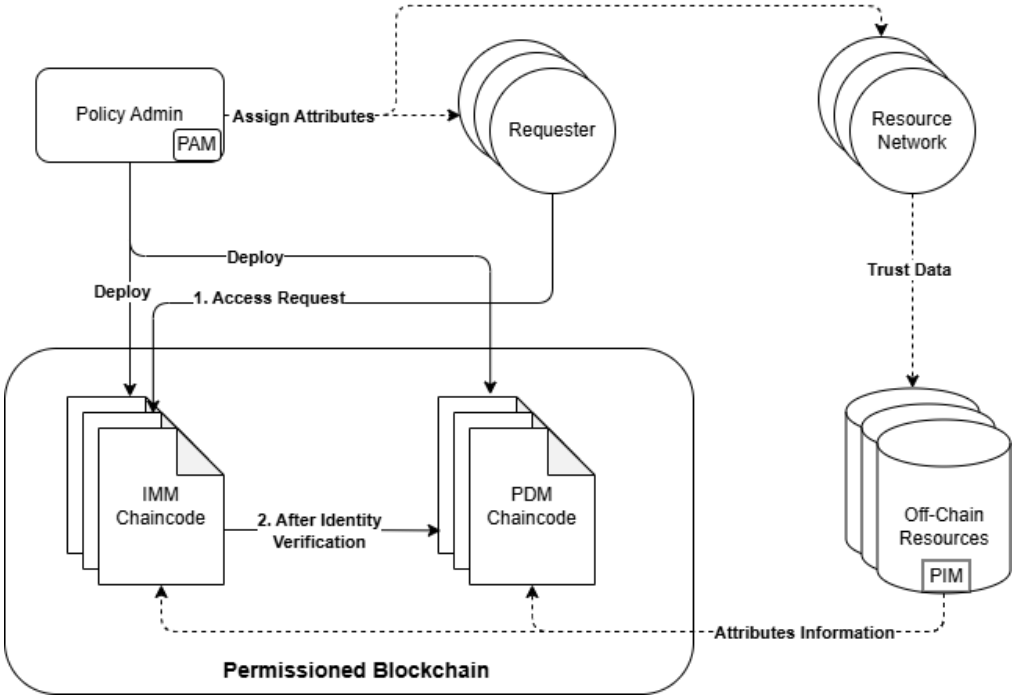

Figure (5.4) Chain codes Deployment on the HLF

decisions rendered by the model, accompanied by immutable audit trails.

Policies for Chaincodes

As shown in Algorithms 1 and 2 of the IMM and PDM chaincodes, there are different policies that are checked and verified to provide access to the resources. For example, in the CheckAccess function of the IMM chaincode, there is a set of conditions that verifies the attributes of a requester against the set rules in a policy. We discuss these rules in more detail to understand the working of a CheckAccess function. These rules use the same attributes as those defined in algorithms.

Rule 1: Authorized Role Validation: It ensures that users with approved roles can initiate the FL process.

$Matches -- > Policy.RequesterRole == User.Role$

IF User.Role ∈ { "Data Scientist", "AI-Researcher", "FL-Coordinator" }

Then Continue

ELSE Access Denied

Rule 2: Data Type Authorization: Checks the data type used to train the FL model.

$Matches -- > Policy.RequestedDataType == RequestedDataType$

IF RequestedDataType ∈ { "Imaging Data", "Non-Confidential Medical Data", "Cancer Patient Records" }

Then Continue

ELSE Access Denied

Rule 3: FL Model Verification: Ensures that the requested model type for the FL matches the approved training models.

$Matches -- > Policy.ModelType == ModelType$

IF ModelType ∈ { "Horizontal FL", "Vertical FL", "Federated Transfer Learning" }

Then Continue

ELSE Access Denied

Rule 4: Ensures that only devices with sufficient trust values are selected in the FL training process.

$Matches -- > Policy.TrustValue <= Node.TrustValue$

Rule 5: Ensures that the requester security requirements meet the security standards set in the policy (e.g., HIPAA-compliant encrypted transfer).

$Matches -- > Policy.SecurityType == SecuritType$

IF SecurityType == { "Confidential" AND Network.Compliance == "HIPAA-Compliant" }

Then Continue

ELSE Access Denied

In the same manner, we define the policies for the PDM chain code shown in Algorithm 2, We discuss some important rules for the PDM chain code that checks the requester and resource attributes.

5.4.5 Access Control Data Flow

Upon receiving a request from a model requester for a trained model from the FL resources, the IMM authenticates the requester's identity and authorizes the request based on the provided characteristics. The Policy Enforcement Module (PEM) employs attribute information from the Policy Information Module (PIM) to generate an access control request. This request will then be forwarded to the Policy Decision Module (PDM) to render policy judgments. PDM will issue a request for information to both PAM and PIM. PAM supplies details on the PDP access authorization policy. PIP consolidates the data obtained from TMM and the attribute database. PDM evaluates the access request's permission or rejection by examining the response from PAM and PIM, and communicates the policy decision to PEM. The PEM renders a judgment and offers feedback to the model requester.

5.5 Trust Management to Detect Malicious Nodes

The selection of a federated learning resource network can profoundly influence the training outcomes. The stability of the nodes within the FL resource network influences their engagement in FL tasks and the caliber of their output. Consequently, a suitable node selection methodology is required to identify more dependable and trustworthy nodes among associated nodes for federated learning activities. A node trust update method may be developed by identifying attributes that signify the quality of the node's performance, like model prediction accuracy, model training duration, or node response time. Nodes with higher trust levels are selected for the federated learning task. This approach is

simple to construct; but, devising a trust value updating process that accurately reflects node dependability and forecasts future performance is intricate.

Calculating trust values involves observing a node's behavior over a lengthy period of time to assess its history actions and anticipate its future performance. The value of the node is determined by its performance in each FL task. The trust of the nodes will be affected by three factors, quality of the trained model q_i by any node N_i in the resource pool, past performance behavior b_i, and node's reputation R_i in FL task.

Table 5.4 presents all the notions used in the trust calculation equations.

Table (5.4) Summary of Notions

Notion	Meaning
q_i	model's quality of a node i
k	k is the iteration number running
q_i^k	model's quality of a node i at $k-th$ iteration
b_i	behavior of node a node i
R_i	reputation of a node i
T_i^k	model update time of a node i at $k-th$ iteration
δ_i^k	normalized delay of node i at $k-th$ iteration
N_i	node i participating in the FL task
b_i^k	behavior of node i at $k-th$ iteration
NQ_i	list of model's quality values for all participating nodes
D_i^k	delay factor of node i at k-th iteration
η_i^k	learning rate of node i at k-th iteration
M	total number of nodes participating in a FL task
T	total number of iterations to complete one FL task
BR_i	list of behavior values for all participating nodes
$Trust_i$	Trust value of a node i

After k times iterations, the local model of node N_i is m_i^k and the model quality result after k times iterations is q_i^k. When the FL process is over, each FL participating node N_i holds the average model quality after k-iterations and will get a model quality evaluation list NQ_i for all participating nodes.

$$q_i = \frac{\sum_1^T q_i^k}{T} \quad k \in [1, T]$$

$$NQ_i =< q_i > \quad i \in [1, M]$$

The behavior of any node i after k times iterations can be expressed as b_i^k.

$$b_i^k =< N_i, D_i^k, tan(\eta_i^k) > \quad k \in [1, T]$$

The equation above shows the formulation of N_i, where, D^k represents the delay factor in uploading the local model, and η_i^k is the learning rate when node N_i participates in the FL task for the k-th time. We further explore these variables in the following equation to understand their affect and how they are formulated.

For the delay factor D_i^k:

Normalizing delay:

$$\delta_i^k = min(1, \frac{T_i^k}{T_{max}})$$

To calculate the delay factor:

$$D_i^{(k)} = \begin{cases} \exp\big(-\gamma_D(\delta_i^{(k)})^p\big), & T_i^{(k)} \leq T_{\max}, \\ 0, & T_i^{(k)} > T_{\max} \end{cases} \tag{5.3}$$

with $\gamma D \succ 0, p \geq 1$

To calculate $Tmax$ for round k, the delays of all 50 nodes be:

$T_1^k, T_2^k, T_3^k, ...T_{50}^k$

We applied the percentile-based setting:

$$Tmax^k = T_{(q90)}^k + M$$

Where:

$T_{(q90)}^k$ = 90th percentile delay, eliminated the extreme outliers

M = safety margin (30% of $T_{(q90)}^k$)

Now we calculate the learning rate LR_i

For the Learning Rate LR_i^k

For better detection, we first calculate the Median Absolute Deviation (MAD) that is not affected by the malicious nodes.

For any iteration k in an FL process, the learning rates of all 50 nodes

$$\eta_1^k, \eta_2^k, ...\eta_{50}^k$$

To calculate the median value: η_{med}^k

$$\eta_{med}^k = median(\eta_1^k, \eta_2^k, ...\eta_{50}^k)$$

then the MAD is

$$MAD_\eta^k = median(\mid \eta_i^k - \eta_{med}^k) \mid$$

Now we calculate the learning rate-based factor

Deviation from median:

$$\xi_i^k = \frac{\eta_i^k - \eta_{med}^k}{MAD_\eta^k + \epsilon}$$

$LR_i^{range} = e^{-\gamma L \xi_i^2}$

For the stability of learning rate

$$\Delta_i^k = \mid log\eta_i^k - log\eta_i^{k-1} \mid$$

$$LR_i^{stable} = e^{-\gamma S \Delta_i i^2}$$

Here $-\gamma L$ and $-\gamma S$ determine how strongly the FL trust model penalizes abnormal or malicious learning rate behavior. We set $-\gamma L = 3.9$ and $-\gamma S = 2.5$.

Now, overall learning rate factor for a node at any iteration k

$$LR_i^k = LR_i^{range}.LR_i^{stable}$$

Calculate the average values for all nodes in k iterations:

$$LR_i = \frac{\sum_1^T LR_i^k}{T} \quad k \in [1, T]$$

In the above equation, LR_i is the average or cumulative learning rate for node N_i after completing T of all iterations.

We can better explain the calculation of behavior value in the formula given below:

$$b_i = LR_i + D_i$$

Now we discuss the third factor, the reputation of the node R_i that affects the trust of the node.

Following each iteration of global model aggregation, the aggregation server will compute the local trust value of participants in that iteration. The third factor in this trust value is the reputation value that we calculated using the SV scheme.

To calculate the trust value of a node N_i we can use the following equation to calculate the trust value of any node N_i in the FL resource pool.

$$Trust_i = \alpha.q_i + \beta.b_i + \gamma.R_i \tag{5.4}$$

To normalize the trust values within the range of [0,1], a sigmoid function can be used that sets all the trust values of the nodes within the range [0.1]. In this way, small or negative values will approach 0 and larger values will approach 1.

$$Trust_i(norm) = \frac{1}{1 + e^-(Trust_i)}$$

In the above equation, $\alpha, \beta, and \gamma$ are the weighting factors to regulate the trust value of node i. We can tune this value to set the weights of the model quality, behavior, and reputation value parameters.

5.6 Evaluation and Results

To validate the accuracy of the ABAC model incorporated with the blockchain, we developed two policy sets for the IMM and PDM modules. The primary goal of combining ABAC with blockchain is to provide precise access control inside the FL network, which is scattered among several healthcare companies. The primary parameter for the analysis

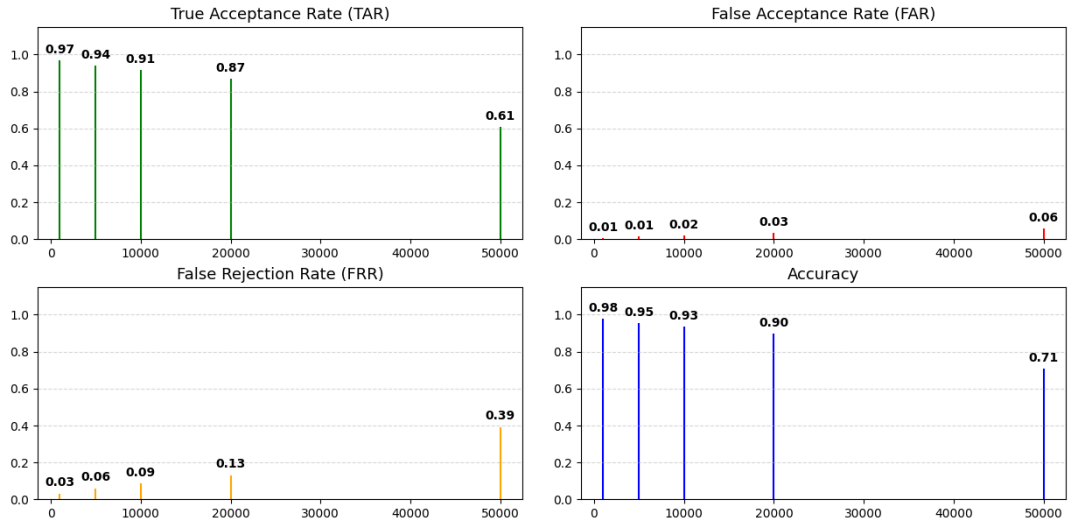

Figure (5.5) ABAC Model Performance for TAR, FAR, FRR, and Accuracy Values

is to assess the accuracy of this ABAC model. We assessed the model's accuracy using a diverse range of queries. The security performance of the ABAC model is assessed using four parameters: True Acceptance Rate (TAR), False Acceptance Rate (FAR), False Rejection Rate (FRR), and overall Accuracy. To accurately align the characteristics with the minimal FAR and FRR values, all attributes in each request are evaluated against the policies established for the IMM and PDM modules. We created 50,000 requests to evaluate the perimeter for the ABAC model, with 30% of the requests incorrect or containing erroneous characteristics. The security efficacy of the ABAC model is illustrated in Figure 5, which shows a range of requests. We illustrated several graphs for TAR, FAR, FRR, and Accuracy in Figure 5.5. The performance of the ABAC model is notably promising across all metrics with a reduced number of queries (1K to 20 K). However, an escalation in the volume of requests resulted in a decrease in performance. In the context of processing 50,000 requests, the True Acceptance Rate (TAR) is 61%, the False Acceptance Rate (FAR) is 6%, the False Rejection Rate (FRR) is 39%, and the total accuracy is 71%. Performance degradations arise from several factors, including network congestion, cache problems, policy lookup delays, and unresponsive peers.

In Figure 5.6, the linear graph representations show performance analysis for the TAR, FAR, FRR, and Accuracy values. The TAR value decreases due to timeouts and dropout requests. An increase in the FRR value is due to the unresponsiveness of the peers, and the accuracy value decreases due to transient failure. We start with the 1000 number of requests, where the TAR and Accuracy values are high. The FRR and FAR values are low at this point. However, by increasing the number of requests beyond 20,000, the overall performance of the system started to decrease. This degrades the performance of the ABAC system and possibly causes delays or inconsistent policy retrieval. Node congestion is the main reason for the degradation, as it increases the transaction latency. In the hyperledger fabric, the PDM chain code is invoked for the access decision, which must pass through endorsing peers, and then the orderer nodes receive transactions to

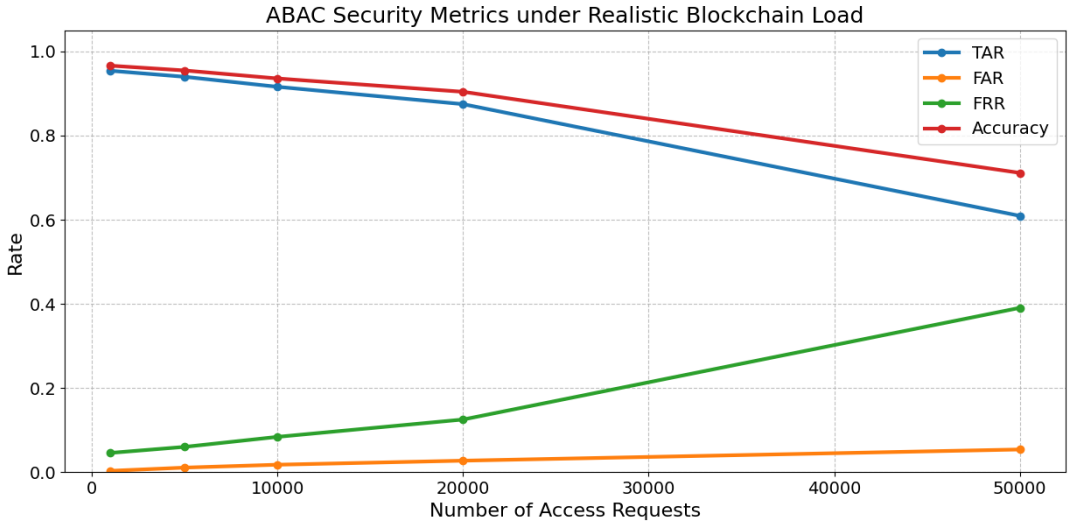

Figure (5.6) Linear Representation of Performance Metrics with Varying Requests

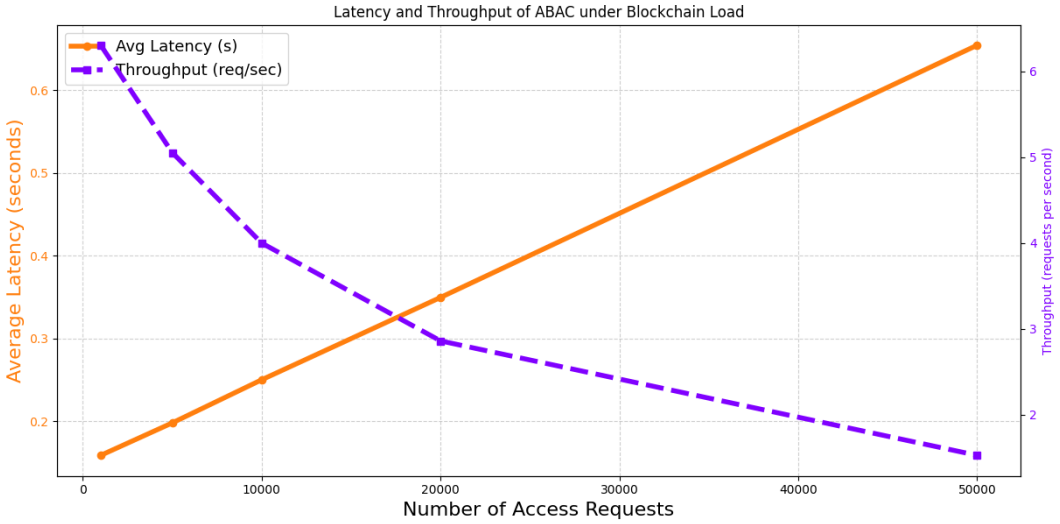

Figure (5.7) Latency and Throughput Analysis with Varying Requests

finalize it.

Figure 5.7 illustrates the average latency and throughput of the system as the number of requests varies. Latency is increasing, while throughput and overall system performance are declining. The outcome is contingent upon the computing capacity of the systems executing the ABAC model on the blockchain and the TMM off-chain.

5.6.1 Trust Management Module Results Analysis

To evaluate the performance of nodes based on their trust values, we set the following experimental setup: When choosing the FL nodes in real model sharing scenarios, a client considers the nodes' reliability as well as their offers. Therefore, this model-sharing method divides all nodes relevant to performing an FL task into two groups based on their trust value. Nodes with a trust value greater than 0.5 are considered high-trust nodes,

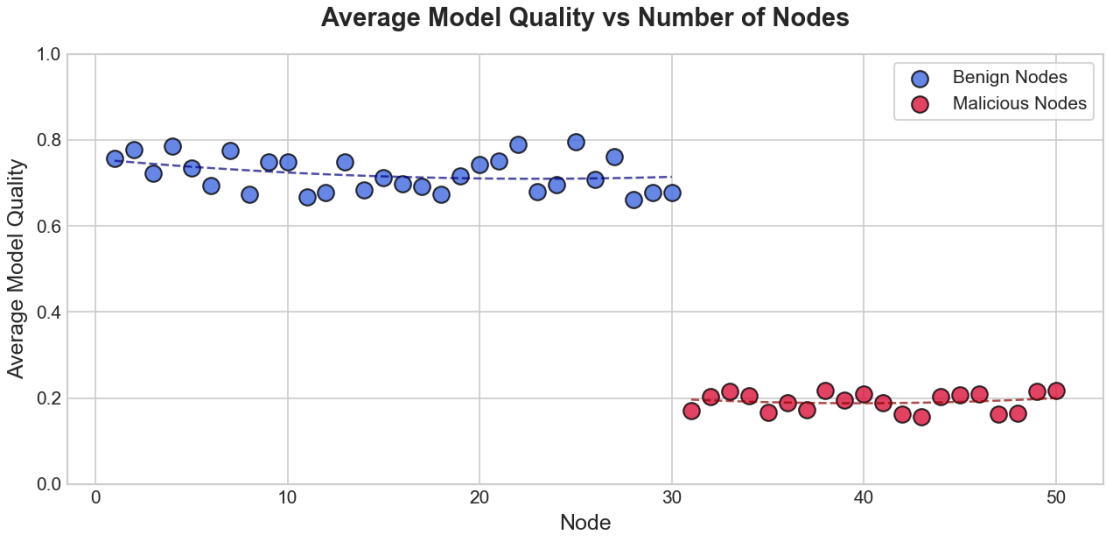

Figure (5.8) Average model quality of 50 nodes for 10 iterations

whereas nodes with a trust value less than 0.5 are considered low-trust nodes. First, we select nodes with high trust values; if the number of high trust values is insufficient, we select some nodes with low trust values.

In the simulation, we initially create 50 nodes and deploy one-third of them as malicious nodes. We set the initial trust values for all nodes at 0.5. The quality of the good nodes model changes between the values of [0.5, 1], and the quality of the malicious nodes changes between the values of [0, 1]. To simplify the experiment, the eagerness of the nodes to become participants in the data model-sharing task is set to 1. The data sharing task is simulated for 50 nodes, where each task is divided into 10 iterations; therefore, there are a total of 5 tasks to share data.

In Figure 5.8, we can see that the average model quality of the 50 nodes is that one-third of the nodes behave maliciously. In this scenario, nodes 1 to 34 behave well, and nodes 35 to 50 behave maliciously. The average model quality of the good nodes for 10 iterations is mainly concentrated in the range of [0.60 - 0.85], which is very close to the real-time situation. The good nodes are plotted in blue, and the malicious nodes are in red color. The average quality of the bad node model is mainly concentrated in the range of [0.15 - 0.25], which we consider to be the bad model quality in federated learning.

In Figure 5.9, average raw delays for 50 nodes are presented, where the average values of actual delays are observed for 10 iterations. It is very close to the real-time scenario, where good nodes share models within the range of [0.8 s to 3.6 s] and malicious nodes take more time, like [7 s to 9 s]. In Figure 5.10, the average delay factors D_i are calculated to normalize the overall effect of the delay factor within the range of [0 - 1]. The good nodes are showing good weights for the delay factor, and the malicious nodes are showing less. These delay factor values are used to calculate the behavior of nodes.

In Figure 5.11, the behavior values of the nodes (Delay Factor + Learning Rate) are the combination of the delay factor and learning rates. This plot helps us to understand the behavior of the participating nodes in the federated learning tasks. The good nodes

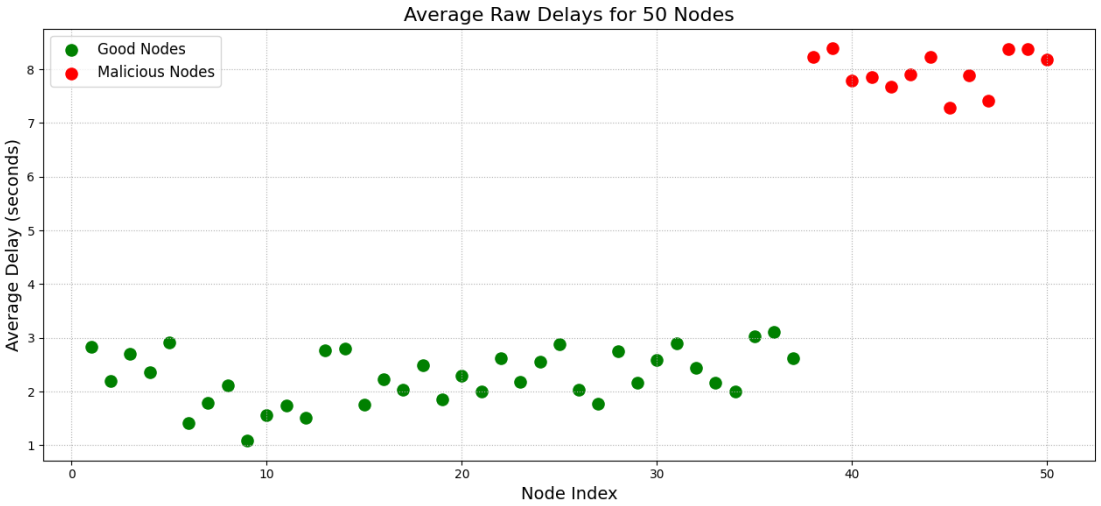

Figure (5.9) Average model delays of 50 nodes for 10 iterations

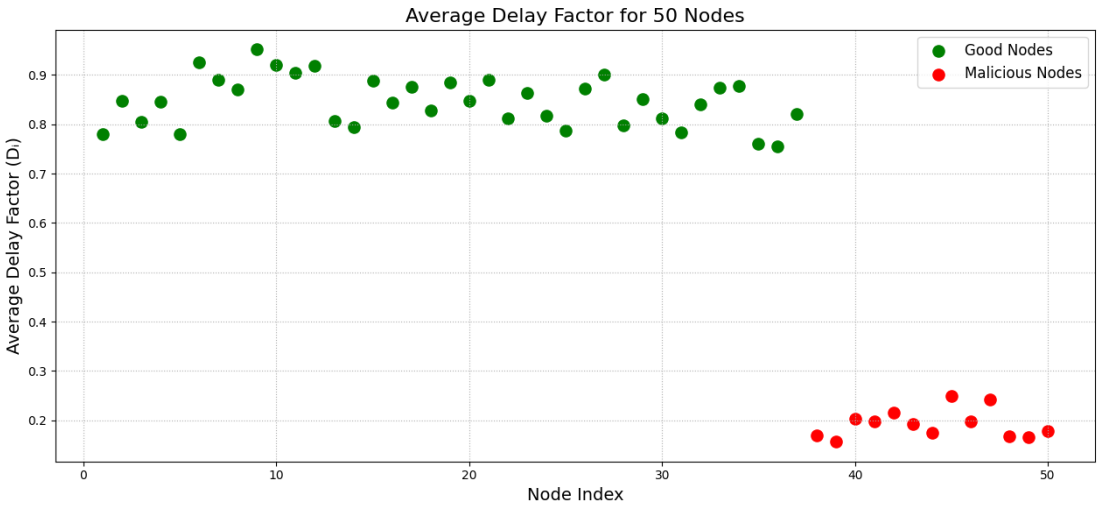

Figure (5.10) Average Delay Factor of 50 nodes for 10 iterations

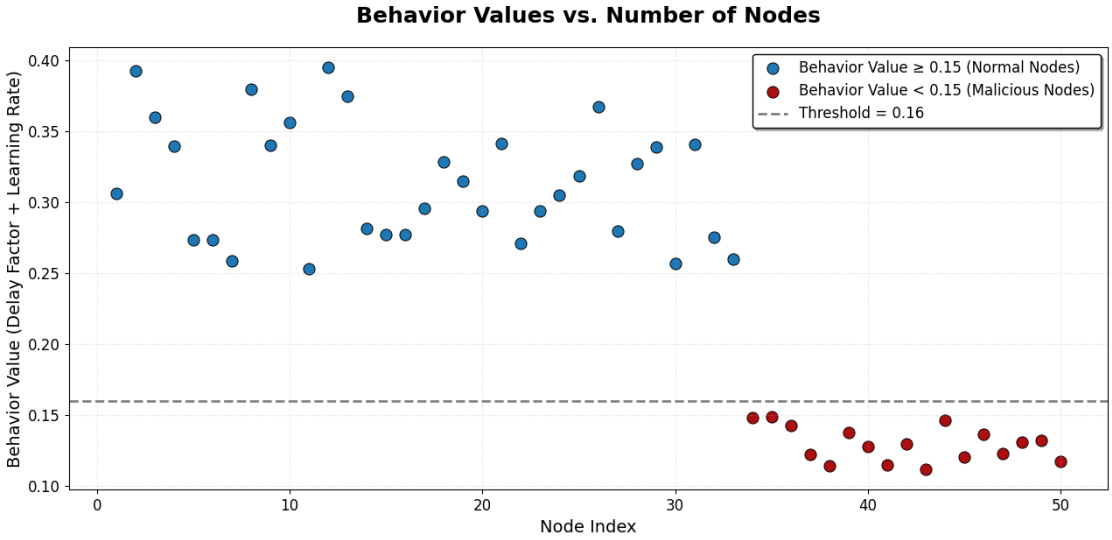

Figure (5.11) Average Behavior Values of 50 Nodes for 10 Iterations

show higher behavior values, whereas the malicious nodes show lower behavior values based on their delay factors and learning rate. Figure 13 shows the average behavior value of 10 iterations for each node. Behavior values for good nodes are above 0.22 and for malicious nodes are below 0.15.

Finally, we compute the trust values for each node based on the quality, behavior value, and reputation value of the model. Reputation values are obtained via the SV method outlined in section 3. The model quality and behavior metrics are subsequently combined with the reputation value to provide trust values, which are utilized to identify hostile nodes inside the federated learning network of any healthcare organization. Section 4 delineates a comprehensive procedure for generating trust values. Figure 5.12 illustrates the progression of trust among 50 participating nodes, with 30% exhibiting malevolent behavior. We conduct 5 federated learning tasks, each executed over 10 iterations, resulting in a cumulative total of 50 iterations for the 5 consecutive tasks. The average trust levels for the malicious nodes diminish as the number of rounds increases. All nodes began their activities with an initial value of 0.5. Nevertheless, the average harmful trust is diminishing. The reputable nodes are excelling, with their average trust value rising due to their model quality, conduct, and reputation values.

Figure 5.13 illustrates that, based on the trust values calculated for both honest and malicious nodes, the ABAC architecture effectively isolates the active harmful nodes. Initially, there are 50 active nodes in total, of which 13 are malevolent. During the initiation of the FL process, malicious nodes are excluded during the initial three rounds or iterations according to the trust levels assessed. The total number of active nodes with favorable trust ratings persists in participating in the federated learning rounds.

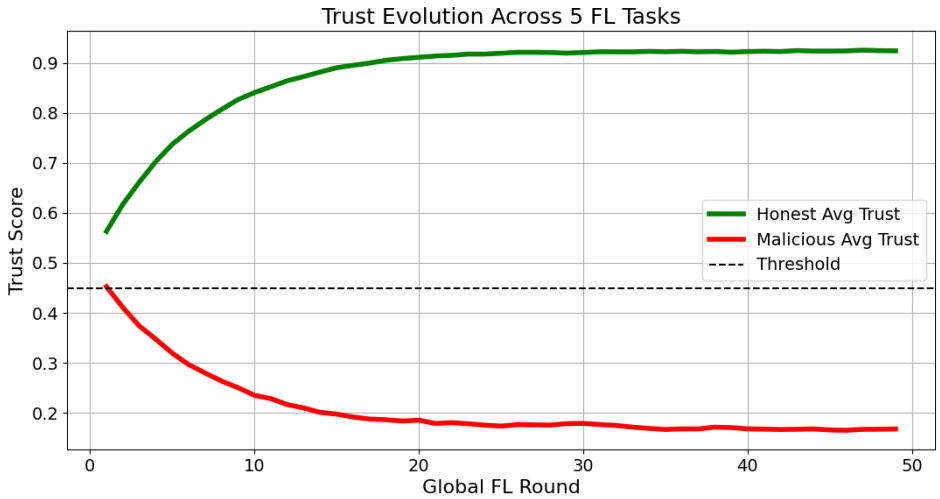

Figure (5.12) Average Trust increasing for good nodes and decreasing for malicious nodes.

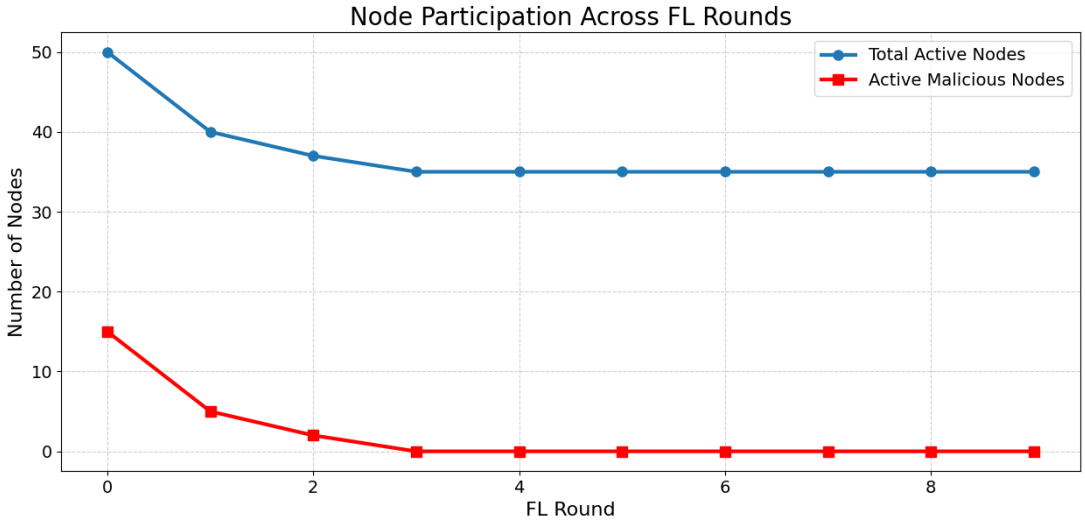

Figure (5.13) Nodes Participation for 10 rounds.

5.7 Conclusion

This chapter addresses the security and privacy problems associated with healthcare data in the realm of data sharing. In the federated learning network, trained models are shared, but the real data remains confidential. However, malevolent or defective nodes might jeopardize the security and privacy of data, which is essential in the healthcare sector. We suggested an attribute-based access control (ABAC) method connected with blockchain to optimize the access control of healthcare data. Furthermore, we incorporated the trust attribute into NIST's defined ABAC model to identify malicious nodes within the FL network and to isolate these nodes based on trust values derived from aggregating the model quality values trained by these nodes, their behaviors, and their reputation values.

Chapter 6

Zero-Knowledge Proofs

This chapter started discussion with the introduction of Zero-Knowledge Proofs (ZKP) and their characteristics. We discuss the types of ZKP systems, their functionalities, and their setup requirements. In addition, major ZKP systems (Groth16, PLONK) are discussed with their working criteria. The second part of this chapter introduces the concept of ZK-Rollup systems, which is essential to discuss for the conclusion of the main thesis problem. We further compare two layer-2 solutions (ZK-Rollup, State Channels) by doing performance evaluation and suggest the best solution for the healthcare data scalability.

6.1 Introduction

Using a peer-to-peer network, cryptographic methods, and a consensus process, blockchain maintains a decentralized, immutable, and verifiable ledger that records all previous transaction data. Transaction data often encompass healthcare information pertaining to a data owner, including medical records, personally identifiable information (PII), and genetic data, all of which pertain to individual privacy. So, security and privacy problems exist.

ZKP is an interactive verification mechanism. This protocol enables the verifier to ascertain the prover's possession of confidential data through a series of predetermined activities, while ensuring the confidentiality of all private information, including that of the prover and the verifier. The verifier is only aware that the prover possesses this data. The execution of this protocol does not need a complex public key, and its repeated application does not assist the malicious user in acquiring further valuable information. Zero-Knowledge Proofs (ZKP) facilitate the execution of anonymous verifiable voting, secure digital asset exchanges, safe remote biometric authentication, and secure auctions.

ZKP must possess these three characteristics to operate effectively;

- Completeness; This property means that the verifier will always be convinced of the provers true statement. Namely, if the prover can prove to the verifier that his statement is true, the verifier always accepts the generated proof.

- Soundness; This property means that the prover cannot convince the verifier that

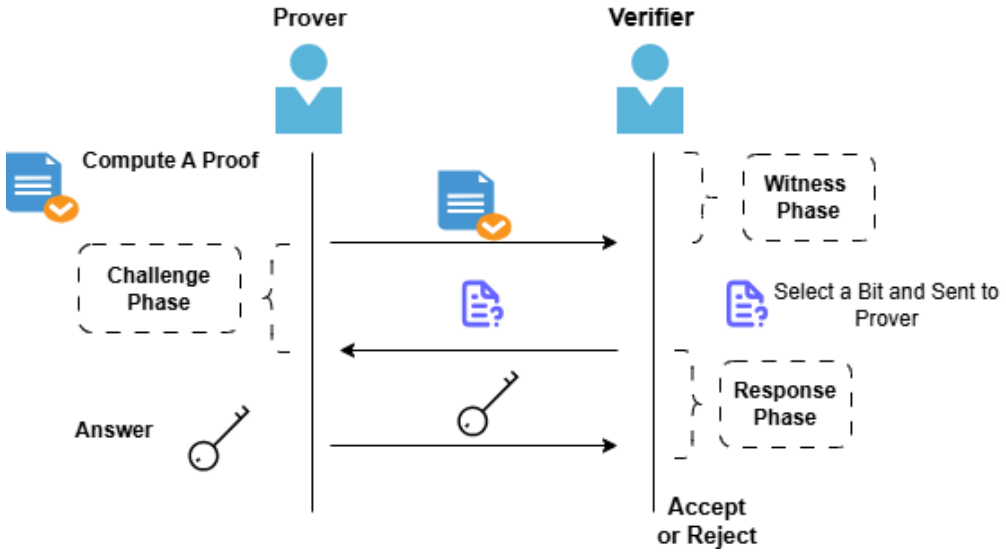

Figure (6.1) ZKP Basic Challenge Response and Proof Verification Steps

his false statement is actually true, except for only a small probability. Namely, if the provers statement is false, the verifier rejects the generated proof

- Zero knowledge; This property can be used to guarantee that the prover does not leak any useful information to the verifier. Namely, if the prover can prove to the verifier that his statement is true, the verifier learns nothing except for the fact that the prover statement is true.

For better comprehension of ZKP and its characteristics, we present an example demonstrating its application in verifying the prover possession of private data. This sample may be utilized for safe authentication within the blockchain ecosystem [51].

In this instance, we assume that the prover has the secret number s. This example's implementation has three steps, detailed as follows:

1. The prover first computes $\nu = s^2 \bmod n$, where $n = pq$, p and q are two large private primes, $\sqrt{n} \leq s \leq n$ is not equal to p or q. Based on the reasonable setup of p and q, the adversary cannot extract the private s from ν, In addition, the prover selects a random integer r, where $1 \leq r \leq n-1$. The prover computes $x = r^2 \bmod n$.x sends it to the verifier.

2. Next, the verifier selects a random bit $\alpha \in \{0, 1\}$. α is sent to the prover.

3. Then, if $\alpha = 0$, the prover sets the proof $\gamma = r$. If $\alpha = 1$, the prover calculates $\gamma = r.s \bmod n$. The prover sends γ to the verifier. Finally, the verifier verifies γ. If $\gamma^2 = x \cdot \nu^\alpha \bmod n$, the verifier accepts γ.

This example satisfies the properties of completeness, soundness, and zero-knowledge, which are described as follows. Completeness can ensure that the prover can provide the correct $\gamma = r$ or $\gamma = rs \bmod n$ to the verifier. Then, the verifier can verify the received γ

and accept it. If the prover does not own the secret number s, soundness can guaranty that the verifier will reject the received γ with the probability 1/2 in each verification process. If γ is verified t times, the probability that the verifier can be fooled is $(1/2)t$. Zero-knowledge means that the verifier only knows ν, x, and γ in each verification process. The verifier cannot obtain prover secret s.

6.2 Types of ZKPs System

6.2.1 Interactive ZKPs

An interactive ZKP system is a more formal type of ZKP where a prover and verifier interact iteratively for the verification of a proof. This multiple interaction ensures fresh randomness from the verifier, and the prover cannot pre-compute the answers. But this ZKP has high network latency due to multiple interactions, and the scalability issue makes it unsuitable for the blockchain.

6.2.2 Non-Interactive ZKPs

Non-Interactive Zero-Knowledge Proof (NIZKP) enables a prover to convince the verifier that a statement is true with a single message, without multiple interactions, and also preserving zero-knowledge property. This feature makes NIZKP important for blockchain, rollups, and decentralized systems. In an interactive ZKP, the verifier sends random challenges, but in NIZKP, the prover generates the challenge itself, embeds it in the proof, and sends one proof that anyone can verify later.

6.3 Transparent and Trusted Setup

Before understanding the transparent and trusted setups in the NIZKP system, it is important to understand the setup in NIZKP. Before generating proofs, it is essential to generate a proving key (pk) and a verification key (vk). These keys are generated with secret randomness; if this randomness is mishandled, then the security of the setup may break.

6.3.1 Transparent Setup

In a transparent setup, there is no secret randomness required. All parameters are derived from hash-based functions and polynomial commitments. All of the security depends on the cryptographic assumptions. As there is no secret randomness then there is no toxic waste exists. In this setup, the proof size is larger and verification process is slower. Transparent setup considered more secure against post-quantum attack but due to their large proof size and slower verification time, not suitable for on-chain. In this setup, ZK-Scalable Transparent Argument of Knowledge (STARK) technique is used to generate proofs.

6.3.2 Trusted Setup

In a trusted setup, there is a one-time initialization phase, where secret randomness is created and then deleted. If the secret (toxic waste) is not deleted, adversaries can use it to generate fake proofs. The participants in the setup generate randomness and then combine it with the system parameters. Then, all participants delete their secret. There are two types of trusted setup: circuit-specific, where a new setup is created for each circuit, and universal setup, where a single setup is used for many circuits. In this research work, we are using a universal setup. In this setup, ZK-SNARK is used to generate proofs.

6.4 ZK-SNARK and ZK-STARK

6.4.1 ZK-SNARK

A protocol ZK-SNARK is a variant of NIZKP designed to integrate non-interactivity between the Prover and Verifier with methods that ensure good performance.

The acronym ZK-SNARK is defined with the following terms:

ZK Zero-Knowledge: The prover does not share information or knowledge about the witness.

S Succinct: The compactness of the proof, which is minimal and can be verified in a sub-linear manner

N Non-Interactive: All communication occurs in a single message.

AR Argument: It says that the correctness property is computationally sound.

K Knowledge: Signifies that the likelihood of a knowledge extractor obtaining a witness is the same as the likelihood of a prover persuading a verifier.

SNARK protocols necessitate smaller proofs, namely less than the witness size, ensuring that the verification time is less than the time required by the Verifier.

6.4.2 ZK-STARK

It is a modification of the ZK-SNARK protocols, established in 2018, distinguished primarily by its enhancement of operational transparency.

Transparency indicates that the above-mentioned preparation procedure is unnecessary, and only minimal cryptographic assumptions are required. This attribute is attained using stochastic methods, as the model employs collision-resistant hash algorithms like SHA-256 or SHA3.

Moreover, they are scalable, engineered to do more extensive computations than SNARKs, and resilient to prospective quantum assaults.

For practical implementation of ZK-STARK, there is a permissionless ZK-Rollup system that is used in the Ethereum network to make it more scalable, cheaper, and secure.

6.4.3 ZK-SNARK Proof Systems

As we have discussed, ZK-SNARK is a class of Non-Interactive Zero-Knowledge Proof (NIZKP). The strengths of this system are that they generate proofs with extremely small sizes, they have mature cryptographic foundations as compared to ZK-STARK, and they are ideal for the blockchain system. There are many systems available to implement ZK-SNARK, such as Groth16, Plonk, and Marlin. However, in this research, we focused on the Groth16 and Plonk ZK-SNARK systems and discussed them in more detail in the upcoming sections.

6.4.4 Groth16

It is one of the most efficient and widely used ZK-SNARK proof generation systems introduced in 2016. Its constant size of proofs and fast verification system make it famous and ideal for the blockchain system. Groth16 is a pairing-based proof generation system that generates a proof almost 192 to 200 Bytes. Moreover, its verification time is very fast. However, it requires a trusted setup specific for each circuit. Groth16 architecture can be discussed with the help of its working criteria in three phases.

1. Phase 1- Setup: It requires a trusted setup where a proving key pk and a verification key vk is generated. In this phase, secret randomness (toxic waste) is generated that must be destroyed after its usage. Then the circuit is created which is specific to a setup. For a new setup, you need a different circuit.

2. Phase 2- Proving: In this phase, there are two inputs: public input and private input. Public input is the information already known to the verifier and explicitly checked during the verification, and it defines what is being proven. Private input is the witness or secret data that satisfies the circuit constraints and is never revealed. The private input is only used by the prover. A prover used the public input, private input, and the circuit to generate a proof π.

3. Phase 3-Verification: On the other side, the verifier checks proof π by using the verification key vk and public input.

A working architecture of Groth16 system is shown in Figure 6.2. For technical understanding of Groth16, to create circuits and generate pk, vk, and proofs, there are two well-known tools available: Circom and SnarkJS. These are two open-source libraries that are widely used to create a setup. Circom is a high-level acDSL used to write zero-knowledge circuits. When writing a Circom code to define a circuit, a developer must define what should be proven, what the public and private inputs are, and what constraints must hold. Circom compiles the high-level language logic into Randk-1 Constraint System (R1CS) that should be handled by the ZK-SNARK prover.

On the other hand, SnarkJS is a javaScript/Node.js libraries that runs the trusted setup ceremonies. It generates a witness, creates ZK-SNARK proofs, and verifies the proof.

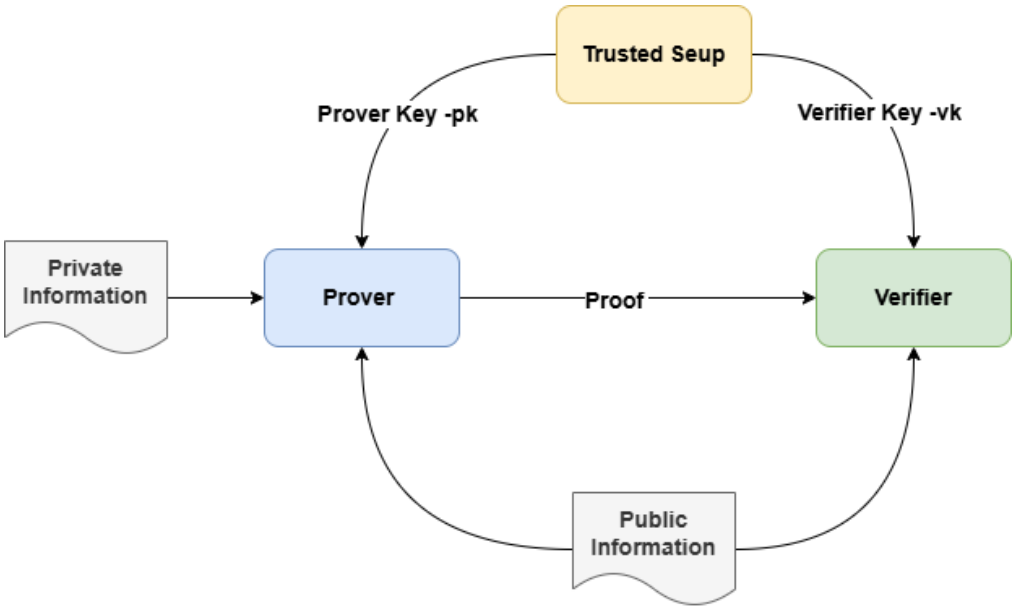

Figure (6.2) Groth16 Working Architecture

A workflow diagram for Groth16 is presented in Figure 6.3

6.4.5 PLONK

Permutations over Lagrange-bases for Oecumenical Non-interactive Knowledge (PLONK) is a more advanced ZK-SNARK system, its architecture is more or less similar to the Groth16 at the higher level but different in using constraints, proofs, and setup environment. It does not require a specific setup for each circuit like Groth16 but can be used in a universal setup, where one created circuit can be used in different setups with different constraints. Like Groth16, PLONK also has three phases of operation.

1. Phase 1- Universal Setup: It requires a universal trusted setup where it generates public cryptographic parameters that all Plonk circuits can use. A single universal ceremony creates the Structured Reference String (SRS) that can support circuits of any size. In Groth16, the circuit wiring is very rigid, but in Plonk it is very flexible. This universal SRS is a public parameter and there is no circuit specific proving key is generated in this phase.

2. Phase 2- Proving: In this phase, public and private input (statement and witness), universal SRS, and circuit description are provided to the prover. The circuit is represented as polynomial identities. Prover fills a table with public and private inputs and commits to witness polynomial with polynomial commitments like KZG. Prover then uses Fiat-Shamir algorithm to create a random challenge derived from a hashing function in a non-interactive way. After satisfying the constraints, it generates a succinct single proof.

3. Phase 3- Verification: On the verifier side, single succinct proof, public inputs, and a verification key that is derived from the SRS and circuit is provided to the verifier.

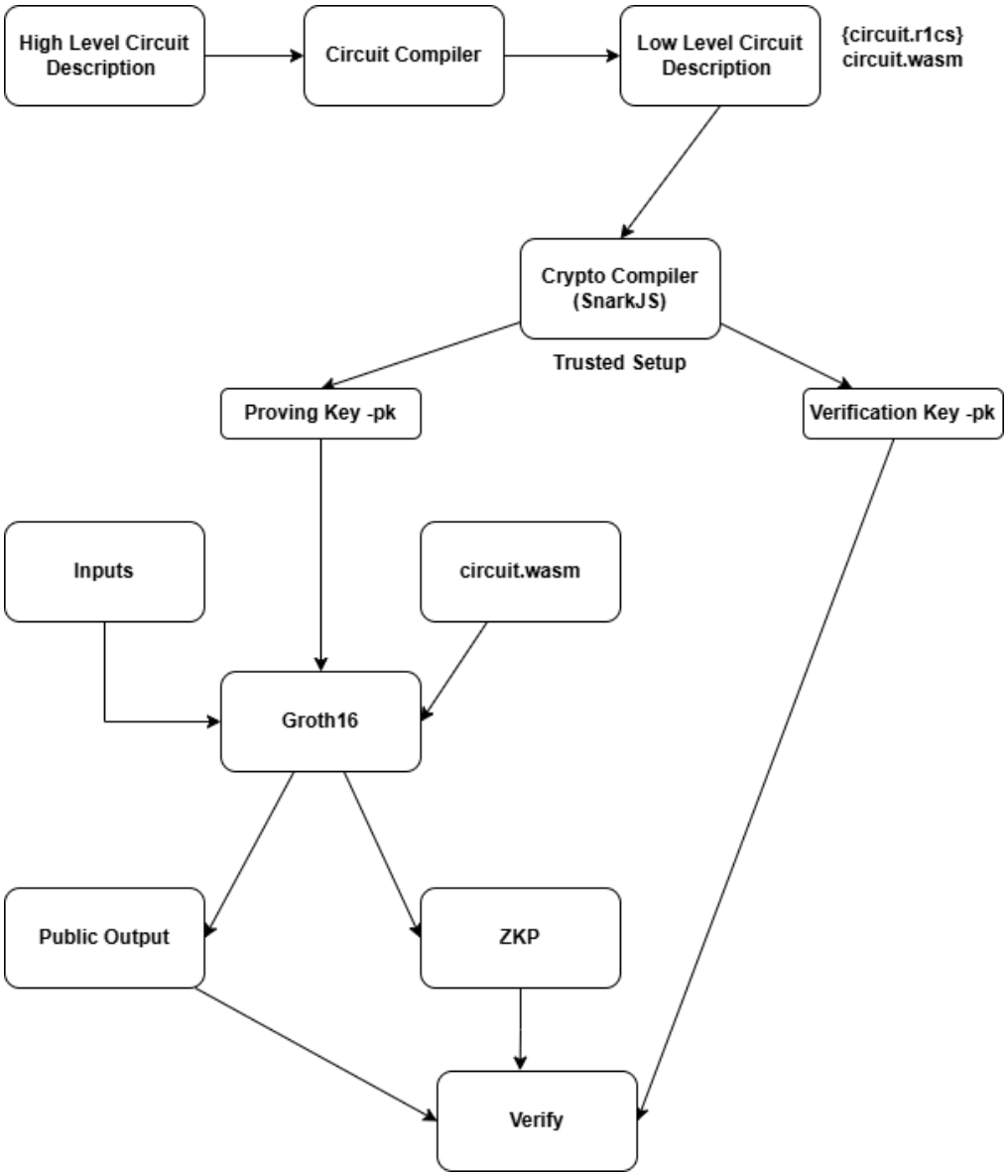

Figure (6.3) Groth16 Workflow

The proof size of the PLONK system is a little higher than the Groth16, but it provides the circuit flexibility, modularity, and up-gradation features. In the healthcare sector, there is a need to evolve for different types of data records and healthcare architecture. Therefore, Plonk is a good choice for such setups. In addition, we need to use the Plonk system with ZK-Rollup, which provides better compatibility with the ZK-Rollup setup.

6.5 ZK-Rollup

ZK-Rollup is a layer-2 (L-2) blockchain system that executes the transaction off-chain and sends a single ZKP to the main blockchain. Provides scalability to the blockchain network where a single proof is forwarded to the blockchain and verified by the smart contract. It ensures security, privacy, and finality. In a ZK-Rollup system, all transactions are batched, and a single proof is generated that proves that all batched transactions are valid. Moreover, it further states that all signatures are correct and there is no double spending. In the healthcare sector, data are highly sensitive and fragmented, and there are strict compliance frameworks (HIPAA, GDPR) that penalize in the case of data leakage or disclosure. Therefore, ZK-Rollup system not only provides scalability to the healthcare system but also ensures privacy because no data is exchanged on-chain and also maintains cryptographic audit trails. A simple ZK-Rollup architecture diagram is shown in Figure 6.4. We can see that users are sharing health records in the form of transactions, and thousands of transactions are handled by the ZK-Rollup system. This system creates a batch of these transactions and then forwards this batch to a ZKP system like Plonk or Groth16 to generate a single proof to validate all of the transactions. Afterwards, this single validity proof is forwarded to the blockchain for verification.

A workflow diagram is presented in Figure 6.5 for the ZK-Rollup system specifically for healthcare patients. All health records are stored off-chain in private databases from hospitals or medical labs. Only data commitments values based on hashing functions or KZG commitments. These commitments also ensure the state updates of the already stored data. These commitments are integrated with the ZKP created by Plonk or any other ZK-SNARK system. Then these ZKP and commitment values in the form of state update is stored on the blockchain.

6.6 Comparative Analysis of ZK-Rollup and State Channels

A consortium-based permissioned blockchain is an ideal tool for ensuring the security of healthcare data. However, protecting privacy and achieving scalability are both crucial and complex endeavors. Different Layer-2 blockchain methodologies for scalability have been examined. This section compares two layer-2 blockchain methodologies (ZK-Rollup and state channels) on throughput, latency, complexity, and manageability within a healthcare data exchange framework.

As healthcare systems advance, Electronic Health Records (EHR) have become the norm for maintaining patient data, medical histories, treatment records, and clinical

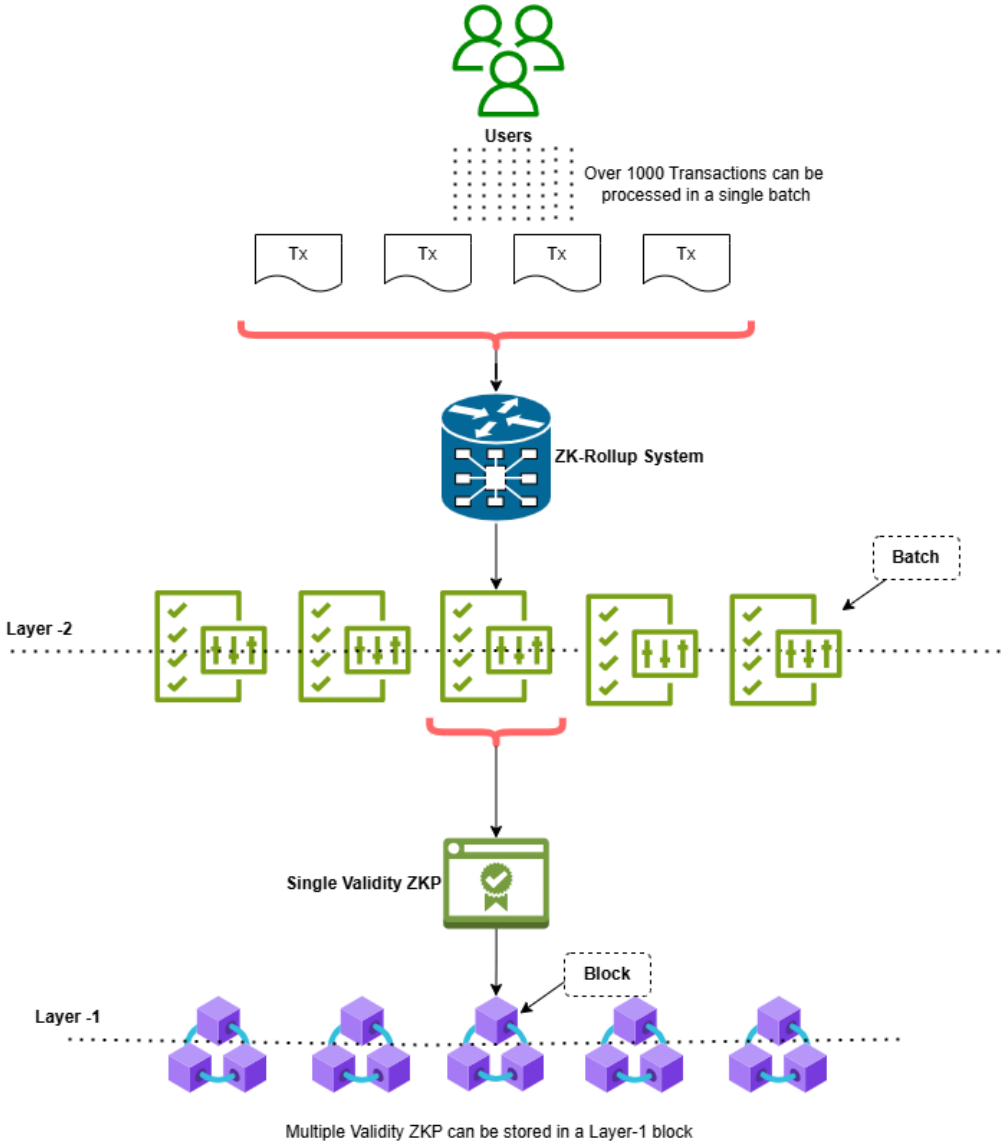

Figure (6.4) ZK-Rollup System

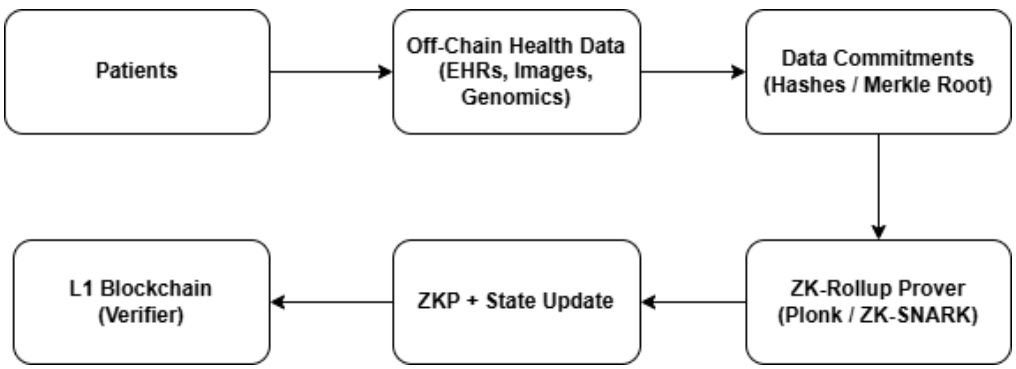

Figure (6.5) ZK-Rollup Workflow

processes [60]. The EHR streamlines data transfer and allows for straightforward retrieval and management by several authorized users. The sensitivity of healthcare data makes it a main target for cyberattacks. The security and privacy measures we employ are effective; nevertheless, they lack the flexibility and scalability necessary to address future assaults, particularly in decentralized systems like Blockchain [42]. However, blockchain systems are not a panacea, as they have their own challenges and limitations.

Since 2019, the adoption of ZK-Rollup (a layer-2 solution) emerged as a viable solution to the scalability challenges associated with blockchain technology. Using zero-knowledge proofs (ZKPs) to provide rapid on-chain verification, enhancing scalability and efficiency while ensuring the privacy of data [10].

The state channel is a further layer-2 option for scalability, enabling the execution of transactions off-chain. Users can perform several off-chain transactions and must submit just two transactions to the blockchain: one for channel activation and another for termination. This mitigates the on-chain verification burden, and hence reducing the amount of on-chain transactions. State channels provide security comparable to authentic on-chain transactions, allowing parties to broadcast their final state on the blockchain at their choosing [82].

This section presents a comparative examination of two layer-2 blockchain scalability solutions through the simulation of a healthcare data exchange model using zero-knowledge proofs (ZKPs). It offers directions for choosing an ideal layer-2 scaling strategy in many scenarios. Furthermore, to our knowledge, no comparison study has previously been conducted on this scalable method.

6.6.1 ZK-Rollup with Healthcare Architecture

In the context of ZK-Rollup, hospitals receive transactions from many healthcare stakeholders, including patients, physicians, and research institutions. Users convey these transactions to a Rollup operator or sequencer system. This operator executes a batch of transactions and produces a single ZK-SNARK proof to authenticate the whole batch of transactions. This singular proof is uploaded to the Layer-1 blockchain along with the state root information. A smart contract run on the blockchain verifies the unique proof and the new global state is affirmed. All data is stored off-chain, and no medical information is sent between users.

As seen in Figure 6.6, healthcare organizations (hospitals, medical laboratories, and research centers) transmit and receive requests for the exchange of healthcare data. Healthcare records are predominantly maintained at hospitals that share medical information with other organizations. For example, when a physician seeks to obtain a patient's medical record for a clinical trial, a request is submitted to the hospital. Following verification of the physician's license or valid credentials via Zero-Knowledge Proofs (ZKPs), a confirmation is provided to the physician that authenticates the patient's medical information without disclosing personal details. This process generates numerous requests, resulting in the creation of a substantial quantity of ZKPs.

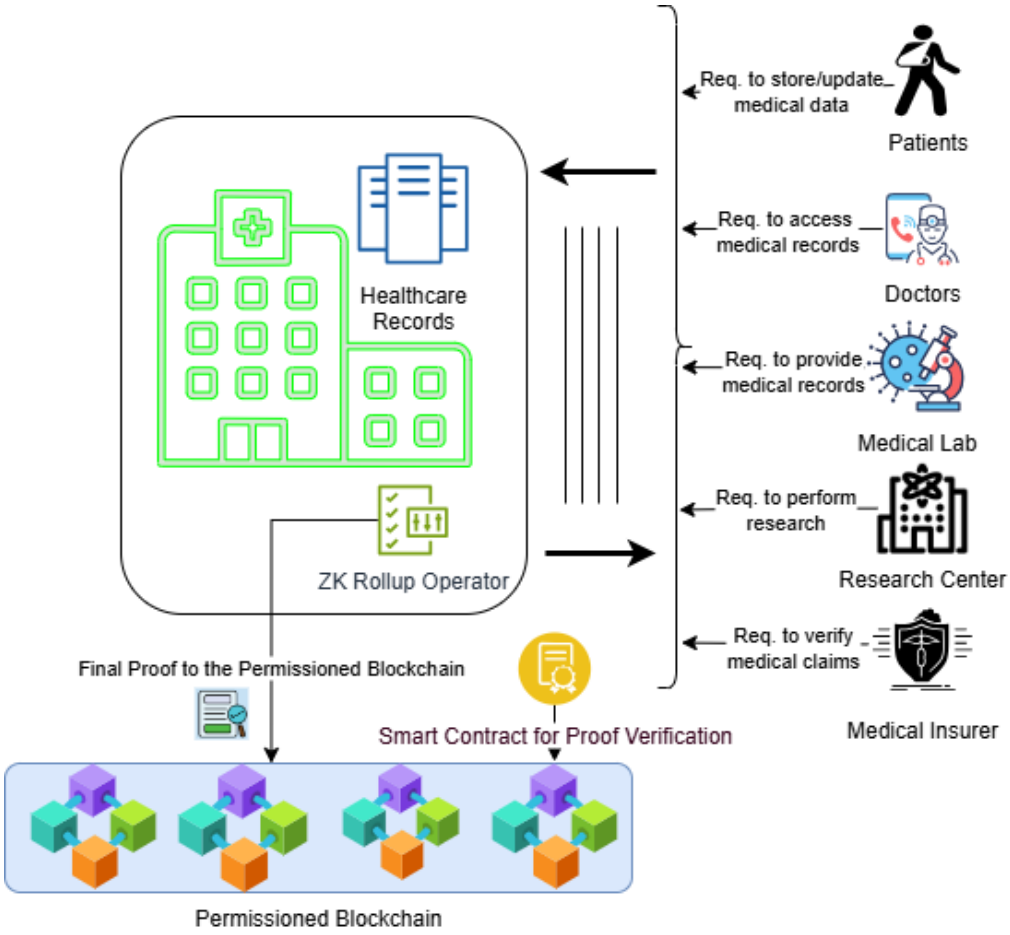

Figure (6.6) ZK-Rollup with Healthcare Architecture

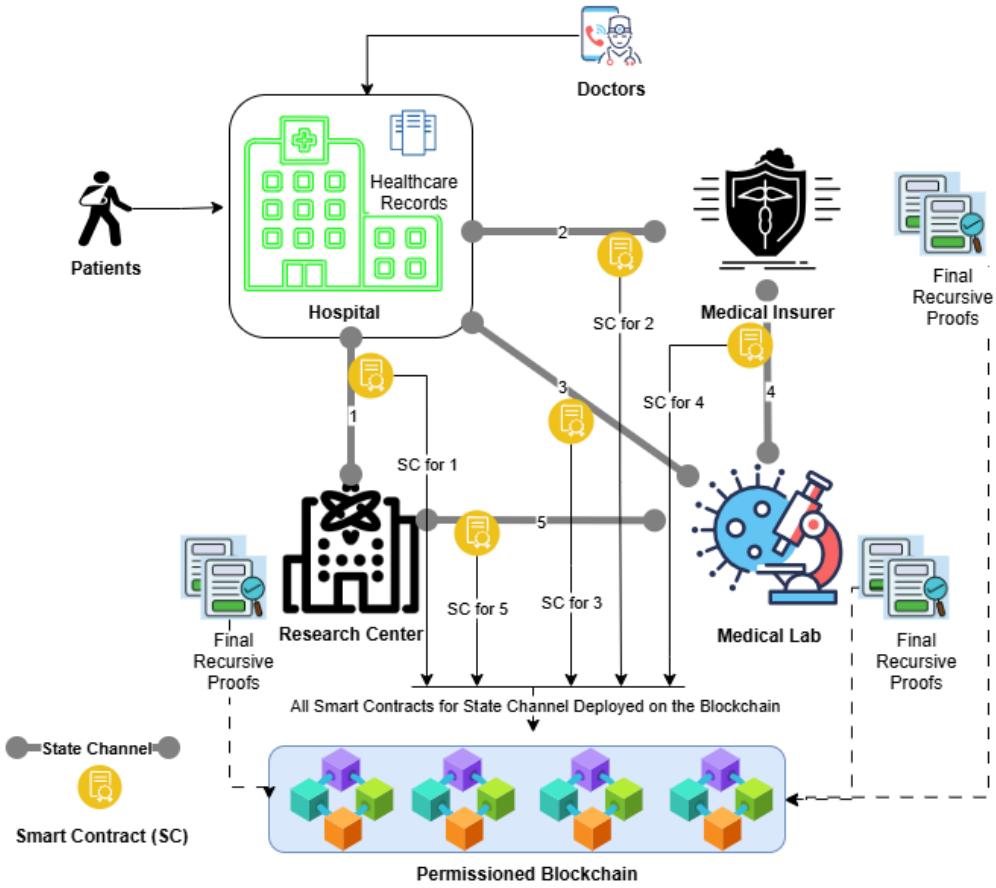

Figure (6.7) State Channels with Healthcare Architecture

6.6.2 State Channels with Healthcare Architecture

A smart contract is used on-chain to establish a state channel by formalizing and securing state conditions between the two parties (e.g., hospitals, research centers, medical laboratories, etc.). A state may be characterized by the fulfillment of specific requirements; for instance, the completion of a series of transactions wherein the requisite information (e.g., laboratory report, patient consent, insurance claim, prescription) is supplied, confirmed, and acknowledged by all pertinent parties. Participants can now exchange signed zero-knowledge proofs with one another over established state channels. Upon completion of the transactions, a concise recursive proof is produced to validate all preceding transactions.

As seen in Figure 6.7, several healthcare companies engage in the interchange of substantial amounts of healthcare data. Instead of medical records, ZKPs are transferred between various entities to verify the accuracy of medical records. This ensures both privacy and regulatory compliance. When a healthcare entity requests data from another entity, ZKPs are exchanged within the state channel (off-chain), and no on-chain transaction occurs until the channel is closed.

6.6.3 Batching and Aggregation

Individuals often confuse these two methods employed to address performance and scalability issues. Batching entails gathering many transactions (proofs or messages) into a single unit before processing later or on-chain submission. Aggregation denotes the compilation of pre-existing proofs and their consolidation into a singular, concise proof for on-chain submission [83]. For ZK-Rollup, we execute the batching operation, but for state channels, we carry out the aggregation procedure.

6.6.4 Healthcare Data Sharing

Healthcare data encompasses several forms of information, including electronic medical records, medical imaging, and genomic data. The amalgamation of IoT (Internet of Things) with wearable devices presents significant issues in safeguarding the privacy of little yet regular data streams.Reference [83]. This proposed architecture omits large images and genomic data because of the necessity for substantial processing resources. The comparison of various layer-2 technologies mostly entails the exchange of digital records in textual form. This scenario has five primary entities (patients, doctors, medical laboratories, research centers, and medical insurance companies) that regularly share data with hospitals. Various objectives exist for the transmission of health information among various entities. We briefly examine these objectives here:

1. Patients: exchanged their medical information with hospitals for diagnosis and clinical trials, require privacy on their personal data.

2. Doctors: make requests to access the patients' medical records to do diagnoses and medical treatment and require privacy on their credentials and patients' data.

3. Medical Lab.: perform medical tests on the patient's specimen by fetching details from patients and hospital.

4. Research Centers: make requests to hospitals to share a bunch of medical records for machine learning model training purposes.

5. Medical Insurers: make requests to hospitals and medical laboratories to provide clinical trials and medical test data to verify insurance claims made by their clients.

6.6.5 Implementation

We employed Kubernetes frameworks to configure the environment, a container orchestration system that automates software distribution. This system provides many services that perform functions like request generation, request reception, healthcare record storage, ZK-Rollup Operator execution, deployment of Hyperledger Fabric (HLF) as a permissioned blockchain architecture, proof aggregation, and state channel management. We may commence all these services by deploying pods in the Kubernetes environment. All of these pods operate on the Linux operating system. The Circom Version 2.0.1 library is utilized

to generate and validate proofs, using the inherent capability to produce proofs using the PLONK and Groth16 methodologies. We used the PLONK methodology to produce the evidence. We used the Incremental Verifiable Computation (IVC)/Folding approach for the aggregation of the obtained proofs, using the NOVA framework [70]. We implemented ZKVM (Zero-Knowledge Virtual Machine) using the R1CS Zero architecture [76] to build the ZK-Rollup operator for broader computational capabilities.

6.6.6 Comparative Analysis

We want to highlight the differences in latency and throughput between ZK-Rollup and state channels to protect the privacy of healthcare data. Moreover, the intricacy and administration of these two layer-2 scalability models are similarly substantial. However, there is no quantitative methodology to demonstrate the complexity and management strategy.

Latency

From a performance perspective, latency must be addressed for these two layer-2 scaling options. Beginning with the ZK-Rollup scaling model, it is shown in Figure 6.6 that many healthcare stakeholders share medical information through the generation of ZKPs. A spectrum of transactions (10^3 - 10^5) was synthetically produced to assess latency. The delay in ZK-Rollup is affected by several factors, such as the duration of batch production by the ZK-Rollup operator, the proof generation period, and the on-chain submission and confirmation times. We utilized the PLONK methodology to produce a unique proof, which is an efficient process for the generation and verification of ZK-SNARK proofs. The magnitude of the generated proof depends on the complexity of the circuit created to produce it, rather than the size of the batch. Figure 6.8 shows the latency of the ZK-Rollup system, wherein the ZK-Rollup Operator consolidates medical data transactions of differing volumes to produce a single proof by batching. The delay metrics for different transaction volumes encompass batch formation time, proof generation time, and proof submission time on the blockchain.

Figure 6.9 depicts the delay of the state channels. The state channels function simultaneously, allowing peer-to-peer interactions among participants. Thus, the concurrent aggregation of proofs, succeeded by the submission of conclusive recursive proofs from each state channel, exhibits significantly reduced latency relative to ZK-Rollup.

Throughput

Figure 6.10 shows the throughput graph for the ZK-Rollup methodology. The throughput demonstrates non-linear characteristics and diminishes as the volume of transactions in the batching process escalates. In ZK-Rollup, throughput depends mainly on the batching process, which results in the creation of a single proof. The duration needed for proof generation and submission is minimal in comparison to the batching process.

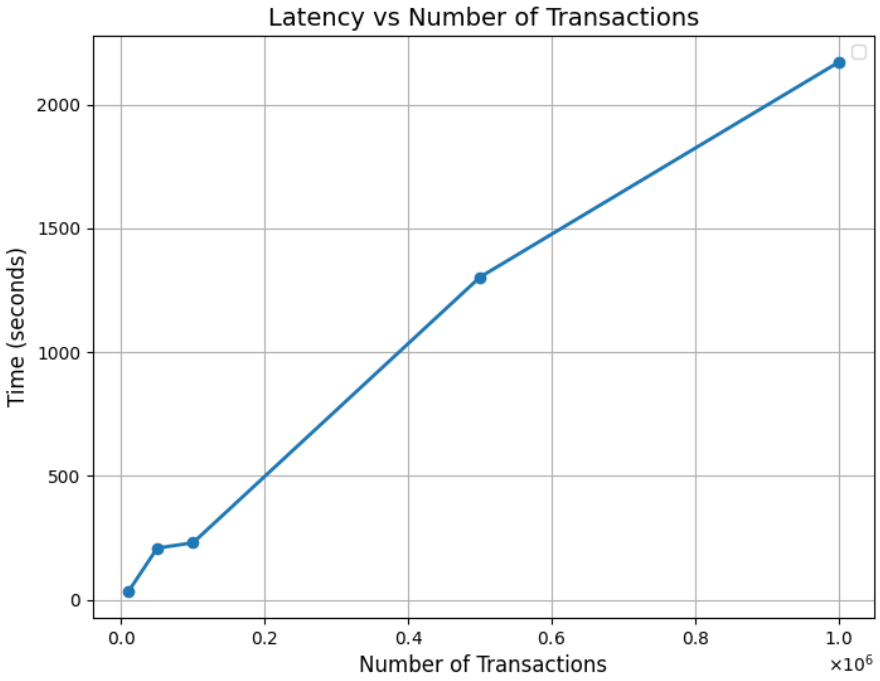

Figure (6.8) ZK Rollup Latency

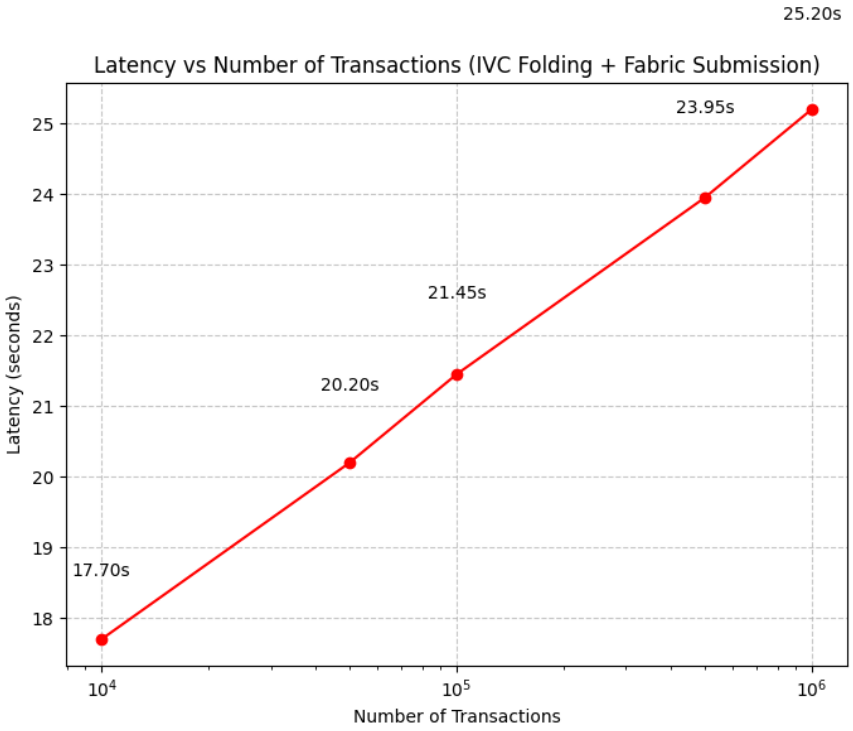

Figure (6.9) State Channel Latency

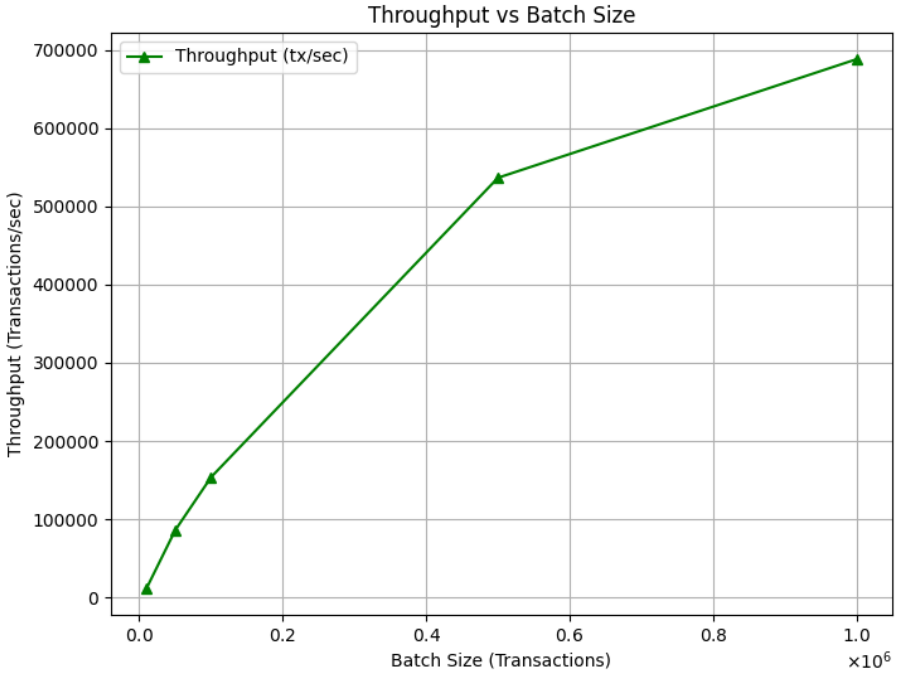

Figure (6.10) ZK-Rollup Throughput

Figure 6.11 shows the throughput of the state channels that exhibit a linear correlation. Diverse pathways are developed among various healthcare institutions. In this case, we are producing requests that are consistently distributed throughout all five channels. Each channel produces a conclusive demonstration using folding-based aggregation. Table 6.1 demonstrates the impact of folds on the throughput of state channels. The NOVA folding approach clarifies the number of folds required to consolidate the evidence for a set of transactions.

Table (6.1) Folding Scheme Affecting Throughput

No. of Trx.	folds = (log2 N)	Folding Time	Throughput Trx./sec
10,000	14	17.50 (sec)	564.17
50,000	16	20.10 (sec)	2,475.25
100,000	17	21.25 (sec)	4,662.00
500,000	19	23.75 (sec)	20,876.83
1000,000	20	25.20 (sec)	39,682.54

6.6.7 Complexity and Management

In terms of complexity, ZK-Rollup systems are more simple to implement than state channels. Each channel is created by implementing a smart contract on the blockchain.

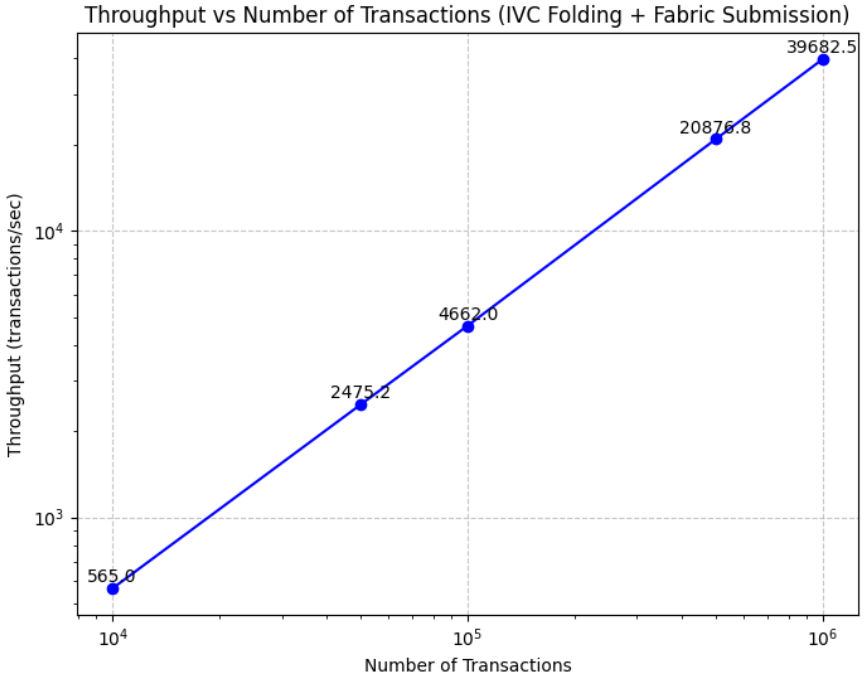

Figure (6.11) State Channels Throughput

Consequently, increasing the number of state channels complicates the generation and deployment of smart contracts. State channels yield effective outcomes with a restricted number of players; nevertheless, an increase in participants escalates the complexity. A state channel constitutes a private connection between two parties. A participant requires a direct route to communicate with all other participants for direct transactions. With 10 participants, 45 state channels must be managed alongside their smart contracts; with 100 participants, 4950 state channels are required. Quadratic growth is unstable due to the necessity of deploying a smart contract for each state channel on the blockchain, with each participant responsible for managing, storing, and monitoring all their channels. This also elevates the system's latency, since an increased number of state channels necessitates the deployment of more smart contracts on the main chain. The ZK-Rollup approach may be globally deployed as it requires substantial processing resources just for the batching procedure, followed by the submission of a single proof to the blockchain for transaction confirmation. State channels provide superior dispute settlement and enhanced tracking options. All transactions transpire off-chain and are immediate; in the case of an incorrect proof, identifying the source of the erroneous evidence is straightforward, as these proofs are verified by the senders. State channels provide immediate finality between participants, as on-chain resolution transpires just during disputes or upon closure. ZK-Rollup encounter delays in finality due to reliance on batching, proof production, and on-chain submission duration. State channels improve the security of healthcare data by only exchanging proofs and aggregating them prior to finalization. Within the framework of ZK-Rollup, transactions are conveyed as proofs and messages that might jeopardize

sensitive medical data.

6.7 Conclusion

In this chapter, we started our discussion with the introduction of zero-knowledge proofs (ZKPs) and explored different types (interactive and non-interactive) of ZKPs. We further discussed two types of Non-Interactive ZKP (ZK-SNARK and ZK-STARK) and focused our discussion on the ZK-SNARK proof system. In the ZK-SNARK proof system, Groth16 and Plonk are discussed with their working architecture. In this research work, we used the PLONK proof system to generate and validate the ZKPs. This chapter also provides insight into the ZK-Rollup Layer-2 blockchain approach that we will use in a case scenario in the next chapter.

This study assesses and compares two layer-2 scaling methods (ZK-Rollup and State Channels) on latency, throughput, complexity, and management in the realm of healthcare data security and privacy. The main contribution is to clarify the notable differences in their functions and effectiveness within the healthcare domain. We conducted this comparison study by modeling a healthcare data-sharing methodology. The comparative analysis section examines the pros and cons of both methods.

Chapter 7

Zero-Knowledge Data Availability (ZK-DA) with ZK-Rollup

The scalability issues of public and private blockchains have limited their use in real-world applications. Layer-2 networks face data availability issues when storing transactions off-chain, especially for non trusted nodes, despite the Ethereum community's progress in ZK-Rollup solutions like the "blob transaction" in EIP-4844. In addition, constructing Layer 2 blocks needs significant computer power, limiting the decentralized nature of Layer 2 networks. This chapter provides novel methods for addressing data availability and decentralization in Layer 2 networks. To guarantee data availability and accuracy, we use "proof of availability" (PoA) to prevent Layer-2 nodes from aggregating transactions without verifying healthcare data.

7.1 Introduction

The popularity of public and private blockchains has grown in recent decades. Scalability has become an major issue for private blockchains as healthcare transaction volumes increase, restricting their use in this area [34].

ZK-Rollup suggest addressing scalability by enabling frequent off-chain transactions (Layer-2, L-2) and storing compressed data on the permissioned blockchain (Layer-1, L-1). ZK-SNARK are used to validate batched transactions. The block size of permissioned blockchain, particularly in Hyperledger Fabric, varies depending on the ordering service managed by the ordering nodes, whereas ZK-Rollup can substantially enhance throughput.

Although ZK-Rollup has potential with Danksharding [93], they present distinct obstacles. The initial concern pertains to data availability, requiring that all transactions remain available. Danksharding [29]introduces two additional issues that impact data availability [25]. Initially, L-1 nodes cannot guarantee the whole data availability of blobs due to the issue of lazy validator. Secondly, previous transactions will be irrevocably destroyed upon the deletion of temporary blobs, resulting in a historical data issue [9]. Although L-2 nodes can retain comprehensive transaction data, off-chain solutions provide additional data availability challenges, including data withholding attacks that impede withdrawals for users, such as patients, physicians, medical personnel, and insurance

firms [20].

The subsequent difficulty is decentralization [12]. Current ZK-Rollup utilize zero-knowledge proofs to demonstrate the appropriate generation of batched transactions; however, the efficient production of ZK-SNARK proofs imposes excessively high hardware requirements on L-2 nodes. Moreover, ZK-Rollup must maintain past transaction records. Consequently, L-2 nodes require increased bandwidth and storage to download and retain prior transactions. Consequently, the L-2 network experiences reduced decentralization due to substantial hardware requirements, resulting in issues such as possible maximal extractable value Maximal Extractable Value (MEV) attacks [102] [62].

This chapter introduces innovative approaches to improve data availability through zero-knowledge and the decentralization of L-2 networks. To address data availability, we initially suggest a "proof of availability" method in which L-2 nodes must furnish evidence of accurate data availability using zero-knowledge proofs (ZKPs) and update states prior to the generation of the subsequent L-2 block. Zero-Knowledge Data Availability (ZK-DA) ensures data availability with commitments, integrity with binding commitments, completeness and correctness with ZKPs. Modern ZKPs systems can handle large data sets with the help of ZK-DA. Moreover, we are introducing a new concept of "PoA" that can make large scale ZK-Rollup systems scalable, fast, and more secure.

Furthermore, we present the PoA to guaranty the accuracy, completeness, accessibility, and integrity of healthcare data within L-2, ensuring high throughput. In contrast to data storage solutions like proof of retrievability (PoR) [2], proof of replication (PoRep) [63], and proof of storage-time (PoSt) [3], our "Proof of Availability" emphasizes data availability rather than storage, offering enhanced efficiency and suitability for L-2 with expedited proving times. We conclude the main contributions of this chapter as follows:

1. Ensuring Data Availability: We present an innovative secure off-chain data architecture to ensure data availability for ZK-Rollup, allowing all participants to access the data. ZKPs alone do not validate that the data is available and downloadable. For this purpose, PoA works by publishing the data to a data availability (DA) layer. Data are split into chunks, and then the sampling (Proof of Availability) process is performed on these chunks at random. Each chunk is checked against the commitment. If data is missing, then sampling fails and that block is rejected. Healthcare data is highly sensitive, large, and frequently updated. So, PoA ensures that these records are available and can be retrieved without revealing the records.

2. Ensuring Data Integrity: Data integrity can be ensured by cryptographic commitments. These commitments can be used as public inputs to the ZKP system. It ensures that data cannot be changed after creating commitments and provides a collision-resistant property. Afterwards, any data that is retrieved must be hashed back to the commitment or otherwise rejected.

3. Ensuring Data Correctness: A ZKP system like PLONK comes into play to ensure the data correctness. In ZK-Rollup, ZKPs provide assurance that a request to

access the data follows specific protocols, signatures are valid, data is present in a specific format like FHIR/LH7, and the old state and new state are valid. If even one transaction or data request is invalid, then constraints of the ZKP system fail, and proof is rejected.

4. Lesser Proving Time: Without the ZK-DA approach, the ZKP system generates proofs on the actual data size. In the case of healthcare data, there are different types of data, like EHRs, imaging, and genomics. Generating proofs on the EHRs may not take much time, but for imaging and genomics data, it requires heavy computational resources to hash raw data bytes, process ordering and encoding, and include large witnesses. This results in large witness files, millions of constraints, and a longer proving time. However, ZK-DA only handles the state-transition logic of the data and commitment values. Raw data is checked with PoA and outside the circuit, which significantly reduces the proving time.

7.2 Preliminaries

7.2.1 KZG Commitments

The KZG (Kate–Zaverucha–Goldberg) [35] commitment scheme is a fixed-size polynomial commitment mechanism that allows a prover to demonstrate the evaluations of a polynomial in designated places. Given that KZG facilitates batch verification for the assessment of many points and polynomials, we utilize KZG as the polynomial commitment in our design to improve efficiency. KZG has three important properties:

1. Succinct: A commitment is a single group element; it does not depend on the size of the polynomial
2. Binding: Once a commitment is created, it is strongly bound to the polynomial. So, no one can change the polynomial after commitment.
3. The value of the polynomial can be proved with a constant-size proof.

The KZG commitments act as a backbone to the PLONK and polynomial based ZK-DA systems [8]. The polynomial encodes the data (transactions batch, FHIR Records, and Rollup State Transition), then KZG requires a trusted setup that generates a trusted value τ. No one should ever know this value. From this value a Structured Reference String (SRS) is published, this is the reason that KZG is not transparent. In the next step, the commitment to the polynomial $f(x)$ is :

$$C = g^{f(\tau)}$$

g is a generator of a elliptic curve group.

This commitment is independent of the polynomial and is of a constant size. Here, the prover does not know τ so it is impossible to create a fake commitment. This ensures the binding and soundness properties of KZG.

7.2.2 FHIR (Fast Healthcare Interoperability Resources)

It is an advanced healthcare data standard presented by HL7 (Health Level Seven International). This standard enables the structured, interoperable and consistent exchange of healthcare information across heterogeneous health information systems [71]. FHIR conceptualizes healthcare data as a compilation of modular reusable resources—such as Patient, Observation, Condition, and Medication—each delineated by a standardized JSON or XML schema with precisely defined data types, terminologies, and interrelations. By using the RESTful API's this standard provides fine-grained access control mechanism for healthcare data while providing compatibility to the conventional healthcare workflows. It is important to discuss that FHIR supports profiling and extensibility, allowing healthcare organizations to adapt standards to local clinical, regulatory, and organizational needs without compromising interoperability.

7.2.3 Sampling (Proof of Availability)

In this scheme, sampling refers to Proof of Availability. It is a mechanism that verifies that data are available and it is integral and correct without doing prover pre-processing. In a ZK-DA design, the prover generates a ZKP over a commitment to the batch data but not on the raw data that a ZK-Rollup system does in its nature. Healthcare data in raw format are published to the DA layer that do not download the gigabytes of data or re-hash the data. Therefore, it checks that full batch data is available for services. So, sampling or PoA mechanism creates a random small number of data pieces on the DA Layer and checks that they are retrievable and consistent with the commitment values. Instead of checking a whole batch of records, it randomly selects a few chunks and compares them with commitment values. If any chunk is missing, the availability fails.

For example, a hospital creates batches of FHIR records (e.g., 128 patients per batch). A batch is split into fixed size chunks.

$chunk - aa$, $chunk - ab$, $chunk - ac$, ...$chunk - nn$,

A commitment is computed over all the chunks by using the KZG commitment scheme.

Commitment $= C = H(chunk - aa||chunk - ab||chunk - ac||...||chunk - nn)$

This created commitment value is used with the ZKP and published on the L-1 blockchain for verification.

Now, A PoA sampler randomly selects k chunks (e.g., $chunk - aa$, $chunk - ag$, $chunk - bg$) and publishes them to the DA layer. Verifies that the chunk exists, and their hashes are checked with the commitment. If all the chunks are available and their integrity persists, then availability is considered valid. The PoA mechanism proves that the data are retrievable, matches the committed batch, and a prover cannot hide records selectively. However, it does not prove the correctness and privacy of the records that we handle with the ZKP.

7.3 Observation and Problem Statement

We evaluated the expenses associated with Layer-2 nodes, focusing on three primary elements: bandwidth, storage and computational costs. Consider an instance in which a 24 MB batch must be produced in 12 seconds. The node requires a bandwidth of 10 MB/s and about 6 TB of storage per month. The costs of bandwidth and storage are substantial, constituting a considerable burden for the L-2 nodes. Moreover, the computational cost is considerable. A batch including 50 transactions using a PLONK circuit necessitates around 225 constraints to provide a proof on 32 vCPUs. The expense of producing a single ZK-SNARK proof for a batch of 50 transactions is considerable for bandwidth, storage, and compute, and will escalate markedly with the addition of more transactions.

The principal issue affecting data availability is the high expense associated with bandwidth and storage. Layer-2 nodes may exhibit problems in downloading and storing batches. Mostly, this is attributable to the expenses associated with bandwidth and storage. In contrast, the problem of decentralization in L-2 networks arises from substantial expenses related to computing. This cost may discourage consumers from participating in L-2 networks.

This study examines the scalability issues of zero-knowledge proofs and decentralization in Layer 2 networks. ZK-Rollup systems have challenges with data availability, including Proving Time and lazy validator difficulties subsequent to the implementation of Danksharding. A possible resolution for the lazy validator issue is to mandate that all L-2 nodes independently download the batch and update their states without relying on L-1, thus ensuring that prior batches remain accessible to other L-2 nodes. Regrettably, L-2 nodes may be discouraged from downloading due to bandwidth expenses. Generating proofs on the raw data chunks with ZK-Rollup also requires additional time. To address these concerns, we implemented the notion of Proof of Availability (PoA), which verifies data availability by establishing commitments for data segments and performing random checks on these commitments to ensure data accessibility. To achieve decentralization, ZK-Rollup requires L-2 nodes to provide ZK-SNARK proofs promptly; hence, it is essential to mitigate the computational demands on L-2 nodes by implementing a Data Availability (DA) layer. Moreover, the data availability layer requires that L-2 publish and retain data transparently, imposing a significant strain on bandwidth and storage resources. Our system requires a reduction in expenses within our design.

This chapter emphasizes data availability rather than data storage. Two scenarios can exemplify the distinction between data availability and data storage. Initially, if any malevolent L-2 nodes accurately record transaction data but withhold them when requested by a challenger. The challenger claims that the necessary transaction data are inaccessible, regardless of its potential storage in L-2. In this case, data storage is guaranteed, although data availability is compromised. Secondly, malevolent L-2 nodes can remove transaction data and subsequently retrieve the destroyed information from other nodes upon request. The challenger asserts that the desired transaction data are accessible but often erased on this node. In this case, data availability is guaranteed, although

data storage is compromised. Whenever data are stored or changed, new commitment values are created on the DA Layer. If the commitment values do not correspond to the randomly produced values during the sampling process or PoA generation, then the PoA is not verified.

7.4 ZKDA Workflow with PoA

If we try to explain the workflow of ZKDA with PoA as presented in Figure 7.1 then a hospital or healthcare entity stores the healthcare records in the FHIR format. These records are stored in the form of Roll-up batches and for the scalability the batch is split into fixed size chunks rather than treating them as raw data record, especially in the case of imaging records.

Afterwards, cryptographic commitments are generated by using the KZG polynomial commitment scheme over all the chunks in the batch. These batches are uniquely bound with the commitments, and to generate the ZKPs, there is no need to process the raw data.

The Rollup sequencer also uploads the data chunks to the DA layer, where they can be retrieved if batch identifiers are known. Here, independent PoA samplers randomly select some chunks from the exposed data to verify their existence and verify them against the published commitments. If any chunk is missing, the PoA is rejected. The cost of sampling is constant, off-chain and independent of batch size, which provides significant results for imaging data.

The ZKP generator with PLONK generates a proof over the commitments that verifies the Rollup state transition is valid and these transitions are mapped to the corresponding batches. A proof with the commitment value is then submitted to the main chain (Hyperledger Fabric) for proof generation.

It is important to share in this workflow model that the DA layer does not forward the PoA to the L-1 layer because sampling is a probabilistic function, not a deterministic one like ZKP. The probability of verifying data availability depends on the sampling rate. More samples are collected from the published data chunks exposed on the DA layer. It only forwards the results of sampling; the data is available or not. If PoA fails, then L-1 also rejects the proof generated for that chunk of data. PoA acts like a watchdog and is not a gate. L-1 only stores the PoA when it is not verified in the DA layer.

7.5 Data Sources and FHIR Format

A part of the dataset was gathered under the SMART BEAR project (Horizon 2020, Grant 857172), which sought to improve healthy and autonomous living for the elderly through personalized treatments. Data include clinical and sensor-derived information from elderly people observed in their home settings by wearable and ambient devices. All data were standardized via the HL7 FHIR framework and organized into five interoperable Parquet tables: patients, encounters, conditions, observations, and questionnaire replies. The

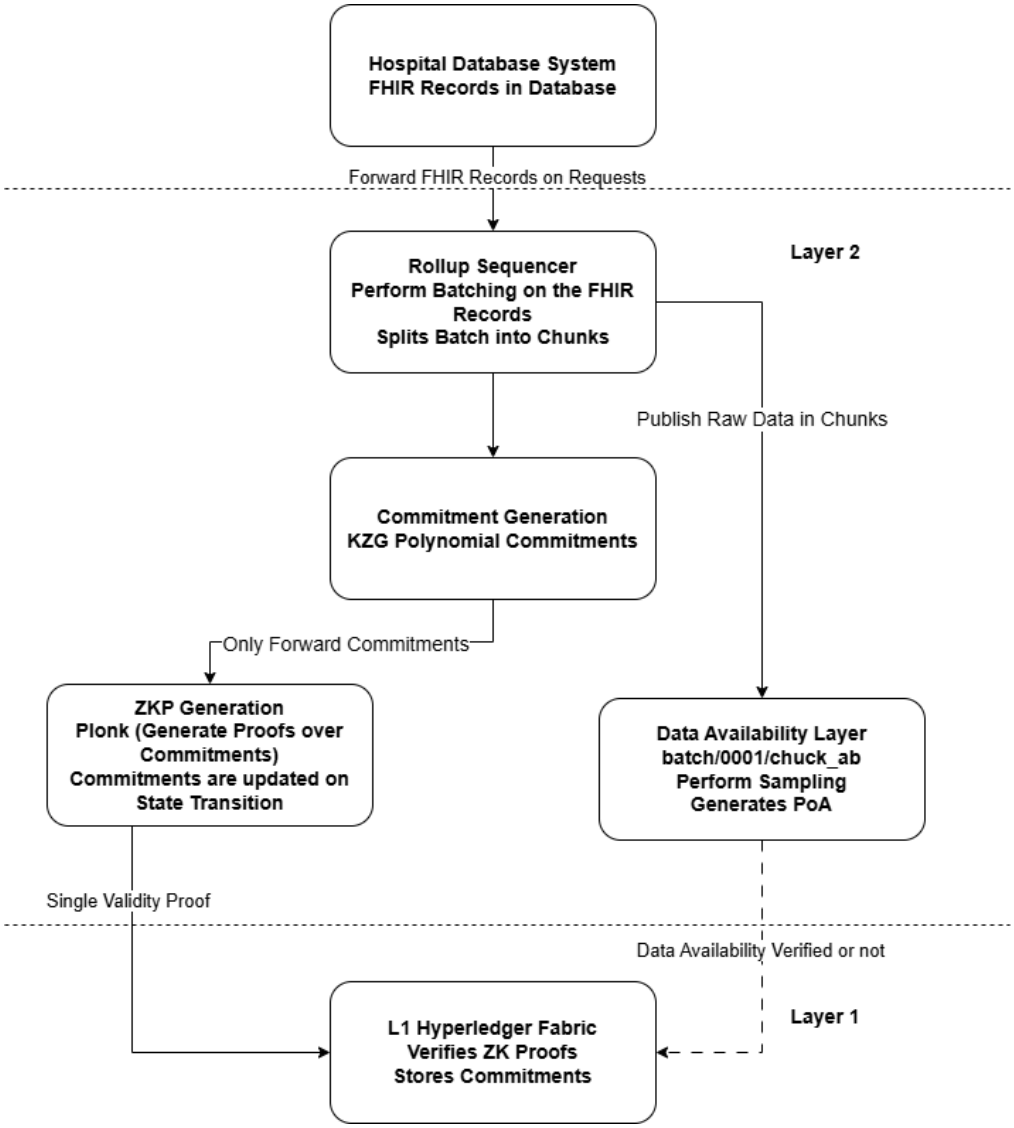

Figure (7.1) ZKDA Workflow

collection comprises anonymized physiological metrics, exercise data, clinical diagnoses, and replies to health-related surveys. The objective is to facilitate research in e-Health, machine learning for clinical forecasting, and evidence-based public health strategies.

The initial data was obtained from an FHIR (Fast Healthcare Interoperability Resources) server. The hierarchical JSON format was converted into relational tables to enable examination in SQL and common data tools.

7.5.1 Anonymization Protocol

To protect patient privacy (Safe Harbor criteria) while preserving longitudinal value, subsequent changes were implemented.

1. Logic: For each of the patients, the first clinical event was identified and shifted to 2024-01-01.
2. Offset Consistency: Every subsequent event for that specific patient was shifted by the same number of days (Offset).
3. Utility: This preserves exact clinical intervals (e.g., days between diagnosis and treatment) while obscuring real-world calendar dates.

This dataset is in Apache Parquet (compressed) format and contains medical records for more than 30,000 unique patient identifiers in EHR, images, and genomics types.

7.5.2 Imaging Data Sources

We need imaging data to check the proving time for this proposed architecture. Because imaging data takes more time to generate the proofs. Generally, imaging datasets are not included in the FHIR format, only meta-data of the images are included in the FHIR records. However, we include imaging data in this study to verify the proving time of the ZKDA + PoA architecture.

For the imaging data, we are using google cloud platform where, a repository with the name IDC (Image Data Commons), where patients datasets are available with different case studies are available online. In this case study, we are using the cancer patient data by integrating these images in the FHIR modules in the observation module.

7.6 Case Scenario

To comprehend the suggested methodology, we constructed a case scenario in which healthcare records are used to sequence, publish, sample, create proofs, aggregate proofs, and ultimately submit a singular validity zero-knowledge proof to the layer-1 blockchain. To have a clearer understanding, it is essential to understand the use of proof. Upon receiving a query from the hospital to obtain a patient's FHIR record, the record is retrieved for publication on the DA layer and for commitment creation. ZK-Rollup systems also execute layer-2 activities, receiving inquiries from other physicians or healthcare entities

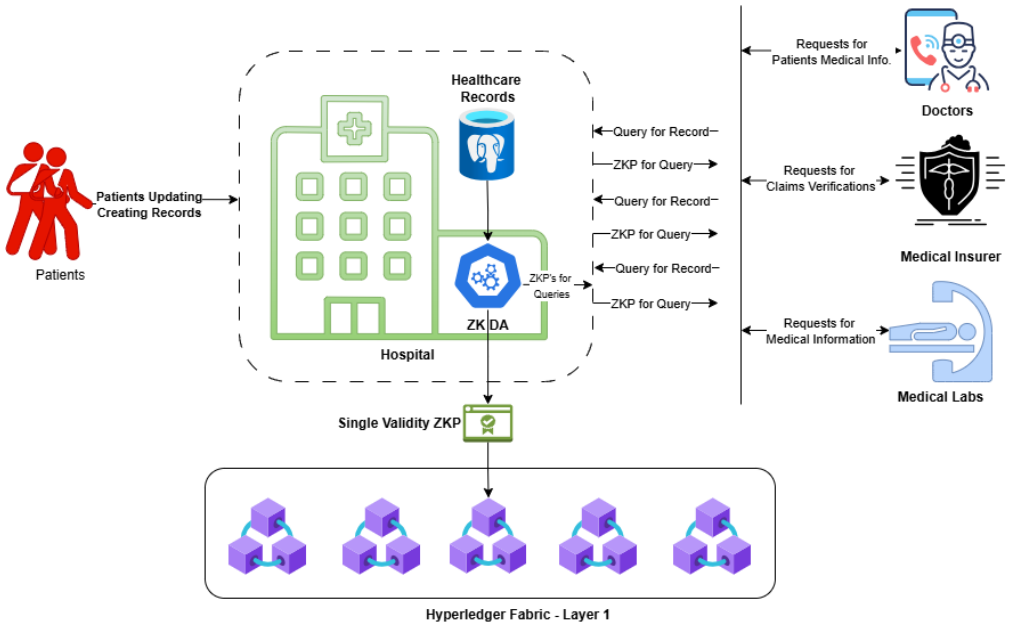

Figure (7.2) ZKDA Data Flow for an Hospital

(hospitals, medical insurers, medical laboratories) to access medical records. To respond to the requested query to access the healthcare record, a ZKP is generated (Query ZKP) and is forwarded to the requester. For example, if a doctor requests to provide hemoglobin information for a patient with patient ID = xyz123 and asks "to prove within a range". If healthcare record of patient ID = xyz123 supports it then the commitment value of that record is used to generate a proof and send that proof to the doctor. If the health record does not support it, then there is no proof for the asked statement. Therefore, the query proofs are not submitted to the L-1 blockchain because the blockchain is optimized for the correctness of state transition but not for the per user queries. Likewise, insurance verification proofs are forwarded to the insurance company integrated with this ZK-Rollup system. However, workflow of the system remains the same. query proofs are generated over the commitment values with the help of PoA. This whole process is executed on Layer-2, therefore, there is no direct involvement of Layer-1 blockchain. The privacy of healthcare data is preserved by not sharing raw medical records with requesters but only proofs of constant size.

Only ZK-DA single validity ZKP are submitted to the blockchain that ensure the correctness of state transition, batch commitment correctness, and new state roots.

In Figure 7.2, a hospital receives requests from different healthcare actors patients, doctors, medical insurer, and medical laboratories to access healthcare records. Patients are registering or updating their medical information using the API gateway. Healthcare records are being updated and stored in the hospital database in encrypted form. Different healthcare actors send queries or requests to obtain medical information from the hospital database for different purposes such as diagnosis, verification of insurance claims, and medical diagnostic lab.

All requests or queries in the form of transactions are processed by the ZK-Rollup

sequencer module, and relevant records are published on the DA layer in the form of chunks are also forwarded for commitment generations and sampling. As shown in Figure 7.1 these commitments are used to generate the proofs. As discussed earlier, now there are two path ways of these proofs, query proofs are forwarded to their relevant requester, and then these proofs are used for the aggregation to generate a single validity proof to submit on the Layer-1 blockchain. Query proofs are stateless and cannot change the state of the blockchain because they do not require consensus. However, the PoA can be forwarded with the query proof to ensure the availability of the data.

In Figure 7.3 an architecture for the case scenario is presented. Different actors (patients, doctors, hospitals, medical laboratories, and medical insurers) are exchanging healthcare information to provide their services. Patients are creating and updating their healthcare records on the hospital database. Doctors are making queries to get access to healthcare records to make diagnoses and prescribe treatments to patients. Medical insurers access healthcare records to verify the claim of patients with health insurance services. The medical laboratories are making queries for the patients observations, and the hospital's results are correct according to the committed FHIR records. The proofs shared with the medical labs and medical insurers are ZK query proofs and verified only on layer-2. These ZK query proofs also provide authenticity and integrity, as the lab results exist in hospital records and have not been altered.

The main points of interaction are hospitals, where most healthcare records are stored and updated more frequently. Therefore, the proposed ZK-DA + PoA method is deployed in a hospital setup. Datasets are stored on a PostgreSQL server [16] in the form of EHR and images, and the sources of these datasets have been discussed in the previous section. Moreover, we are also using some synthetic datasets available in the Kaggle dataset repositories for portable device data.

The ZK-Rollup setup, which has been deployed in hospitals and is equipped with ZKDA and PoA functions, is currently operating to provide scalability and privacy for healthcare data. They are creating batches of transactions and generating query proofs forwarded to the requester and also creating a single Rollup validity ZKP to submit on the layer-1 blockchain. A Rollup smart contract (chaincode) has been deployed on the Hyperledger Fabric (layer-1) that performs the validation process on the submitted single ZKP to verify the correctness of all the batched transactions.

7.6.1 Kubernetes Environment

We are using the Kubernetes framework to deploy a real-time setup that can perform all the operations we have discussed so far. We have discussed the Kubernetes environment in the previous sections as well. Kubernetes offers the ability to establish an environment for the deployment of various services in the form of pods. You can initiate different services in a namespace which functions as a cluster of pods. For illustration, Figure 7.4 shows a namespace environment created in Kubernetes where different services are deployed to run an application. These services are deployed in the form of pods that have

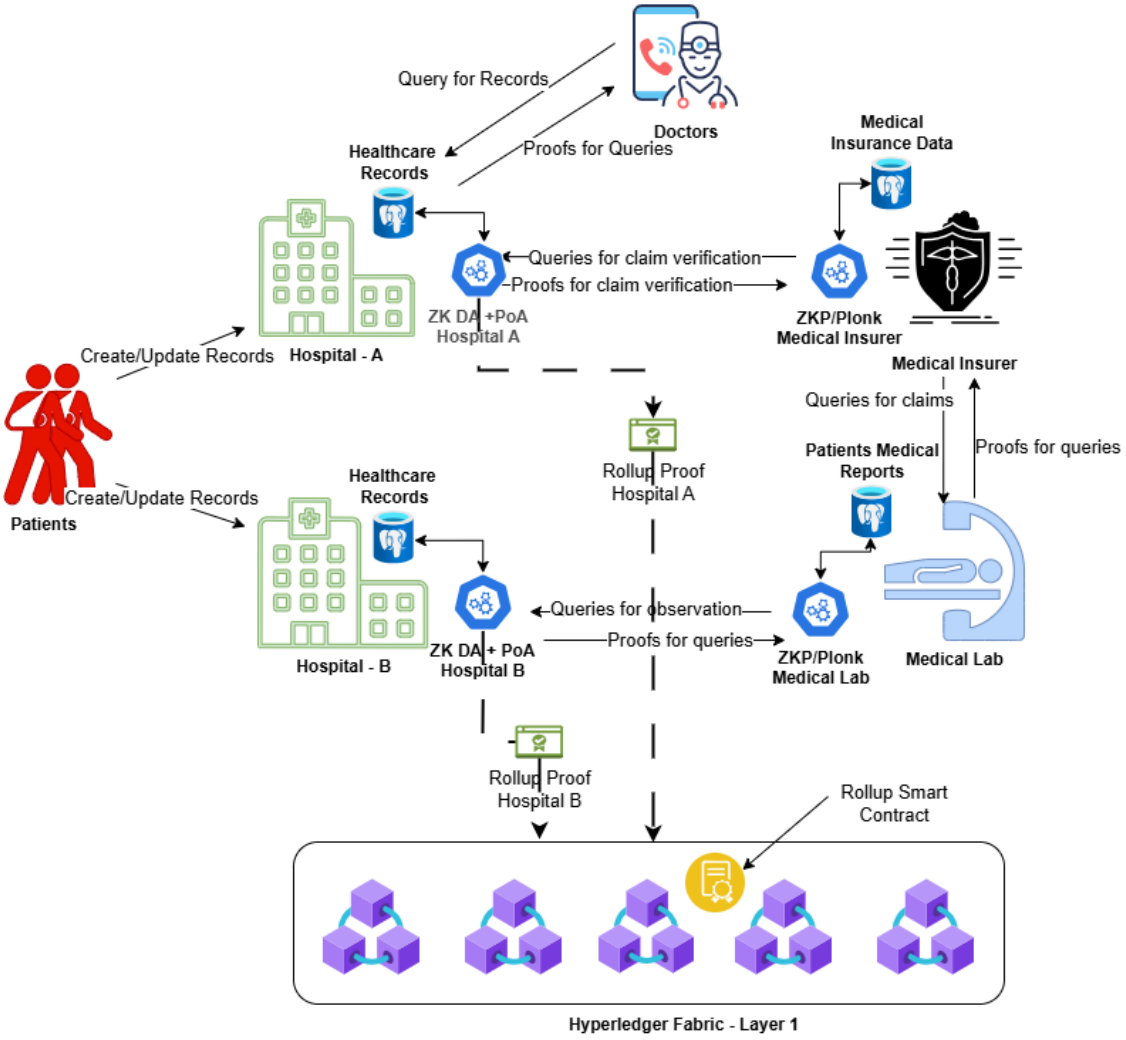

Figure (7.3) ZKDA + PoA and Healthcare Proofs Sharing Architecture

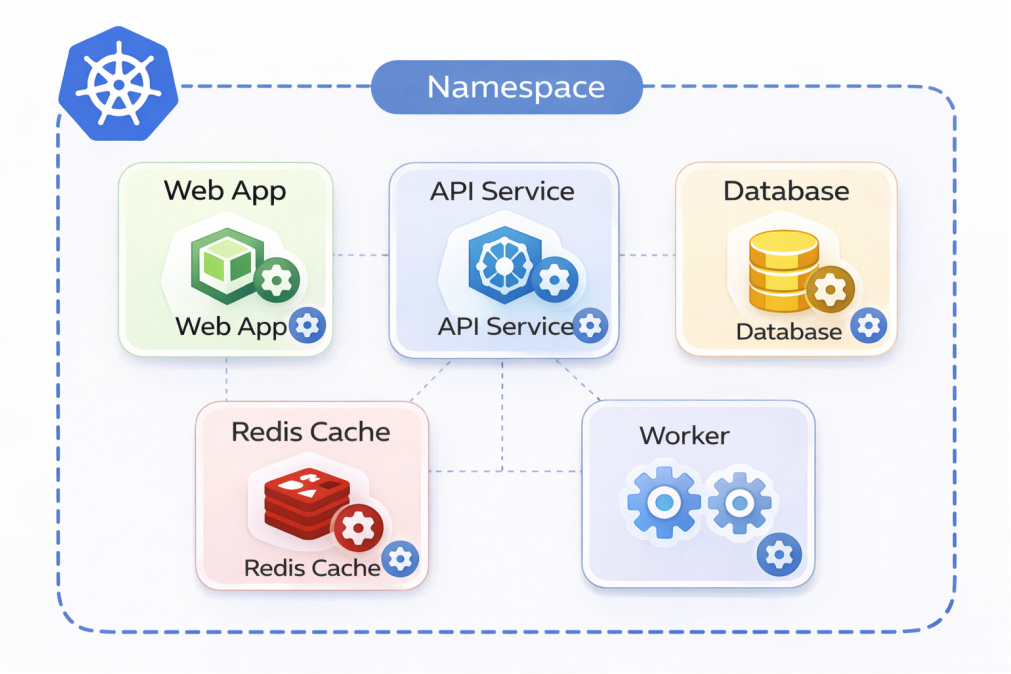

Figure (7.4) Pods deployed in Kubernetes namespace environment

their unique IDs and IPs to communicate with other pods. In the same way, we create different namespaces to run a hospital environment, where the database service, API gateway, ZK-Rollup, ZK-DA and PoA services are running to perform their operations.

To create a namespace for an hospital environment where a database service, with API access to the users is created. In addition, ZK-Rollup with ZK-DA and PoA services are also initiated to perform their operations.

1. postgres-0: Stores medical records in FHIR format (EHR's, Images)

2. sequencer: Receives requests from API gateway and schedule for batching

3. da-publisher: Uploads data chunks and commitments to DA node

4. poa-sampler: Performs PoA sampling, randomly selects chunks from DA node, and records PoA status

5. prover: Generates PLONK proves for the roll-up batches

6. da-node: It provides space to data chunks and commitment files

7. submitter: It takes ZKPs and public input and forwards to the Layer-1 blockchain

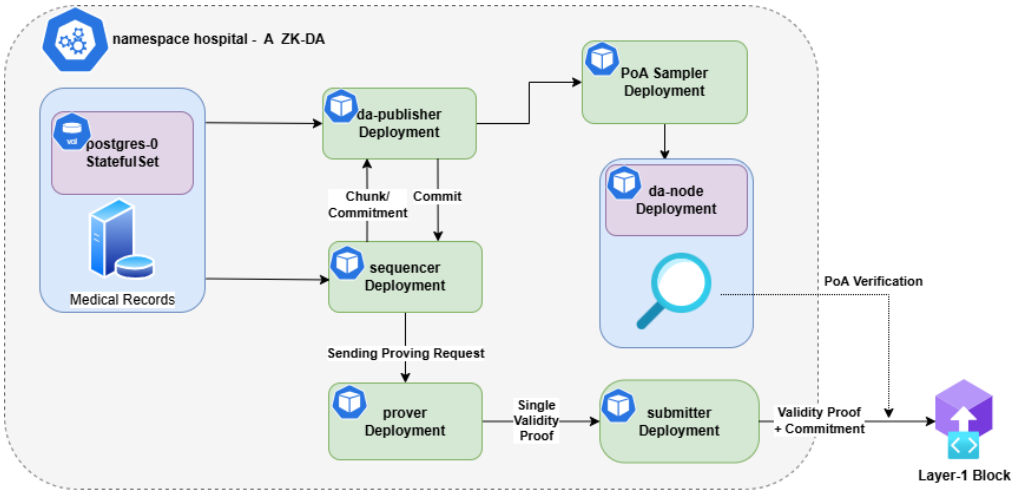

Figure (7.5) ZKDA + PoA Pods deployed in a namespace with their data flow

7.7 Mathematical and Numerical Representation of ZK-DA + PoA with Commitments

7.7.1 Notations and Symbols

Suppose, F be a finite field or a scalar field

and

G_1, G_2 are cyclic groups of prime order p with a bilinear pairing

$$e : G_1, G_2 \to G_T$$

if we say bilinear:

$$e(g_1^a, g_2^b) = (g_1, g_2)^{ab}$$

Now, a structured reference string (SRS) for KZG is:

$$SRS = (g_1, g_1^\tau, g_1^{\tau 2} .... g_1^{\tau d}, g_2, g_2^\tau)$$

secret $\tau \in F$ should be unknown to all parties.

It can provide polynomial commitment up to degree d.

7.7.2 Payload Polynomial Representation

For instance, a Rollup batch contains n transactions / FHIR records update

$$\tau = [t_1, t_2, ...t_n]$$

Now we serialize this batch τ into bytes

$$B = Enc(\tau) \in [0, 1]*$$

Now these serialized batch split into chunks and then map into field elements
Splitting B into N chunks:

$$(c_0, c_1, c_2, ...c_n)$$

mapping each chunk to a field element:

$$m_i = ToField(C_i) \in F$$

$f(X)$ is defined to encode the field values of chunks
f is the polynomial representation of DA payload

$$f(X) = \sum_{i=0}^{N-1} m_i X^i \quad (\text{degree} < N). \tag{7.1}$$

Now, the KZG commitment for the payload

$$C = Commit(f) = g_1^{f(\tau)} \in G_1$$

This is a single compact object that holds the entire batch

7.7.3 Correctness with Commitment

Now, it is better to define how a ZK-Rollup node proof a statement

Assume, S is old rollup state, and S° is a new rollup state. We are applying deterministic state transition:

$$S^\circ = \delta(S, \tau)$$

So, there is a need to identify public and private inputs.
Public Input:

$$x = (S, S^\circ, C, batchID, metadata)$$

Private Input:

$$w = (\tau, B, m_0.....m_{N-1}, f)$$

Now relation $\mathcal{R}$ could be:

$$\mathcal{R}(x, w) = 1 \iff \begin{cases} S^\circ = \delta(S, \mathcal{T}) \\ B = \text{Enc}(\mathcal{T}) \\ \forall i,\ m_i = \text{ToField}(c_i) \text{ from } B \\ f \text{ encodes } (m_0, \ldots, m_{N-1}) \\ C = g_1^{f(\tau)} \\ \text{(optional) consent/policy constraints hold} \end{cases} \tag{7.2}$$

Now the proof generated by the PLONK ZK-SNARK system:

$$\pi \leftarrow Prover(x, w)$$

and

$$Verifier(\pi, x) = 1$$

7.7.4 Proof of Availability with Sampling

The publisher node uploads the raw chunks to the DA layer

$$\forall_i \in [0, ...N-1]$$

DA Layer Store Batch ID, i, c_i
Any chunk can be retrieved with the batch ID

$$c_i \leftarrow DA - Get(BatchID, i)$$

PoA is also an off-chain method and chooses k indices is chosen to collect the samples.

$$I = [i_1, i_2, ...i_k]$$

I is the set of chosen samples to check the availability.
The sampler fetches the chunk by using random indices.

$$c_i \leftarrow DA - Get(BatchID, i)$$

and maps to field element:

$$m_i = ToField(c_i)$$

Now the sampler must verify that m_i is consistent with C, and this is done by the proof opening.
To open f at any point $z \in F$, we define:

$$y = f(z)$$

The KZG opening proof is:

$$\pi_z = g_1^{q(\tau)} \in G_1$$

Now, the verification equation becomes:

$$e(C/g_1^y,\, g_2) \stackrel{?}{=} e\Big(\pi_z,\, g_2^{\tau - z}\Big)\,. \tag{7.3}$$

This must be equal to:

$$e(C, g_2) \stackrel{?}{=} e(\pi_z, g_2^{\tau - z}) \cdot e(g_1^y, g_2). \tag{7.4}$$

If true, then $y = f(z)$ is consistent with the commitment.

Now, the PoA acceptance condition is:

$$\text{PoA}(\mathit{BatchId}) = \bigwedge_{i \in I} [\text{DA} - \text{Get}(\mathit{BatchId}, i) \text{ succeeds} \wedge \text{KZGVerify}(C, z_i, m_i, \pi_{z_i}) = 1]\,. \tag{7.5}$$

If the equation is true, then batch is available with some probability.

7.7.5 Proving Time Reduction

If $|B|$ is the size of the payload (record size), and N number of chunks are created of this payload.

Without ZK-DA + PoA the proving time depends on the number of circuit constraints created while creating a ZKP and the constraints increase with the size of payload.

$$\varsigma_{no-DA} = \Theta(|B|) + \Theta(n) \Rightarrow T_{prove-no-DA} = \Theta(|B|)$$

However, KZG-based ZK-DA, circuit only needs to bind C and enforce state transition.

$$\varsigma_{with-DA} = \Theta(|n|) + \Theta(1) \Rightarrow T_{prove-with-DA} = \Theta(|n|)$$

Therefore, the proving time is independent from the payload size but the constant commitment size.

7.8 Evaluation and Results

We implemented a prototype of our proposed system to demonstrate its feasibility and evaluate the performance of our technique. As discussed earlier, we run in the Kubernetes environment where different pods are initiated to run this prototype. To generate the ZKP for the PLONK system we utilize the Circom version 2.1.6 library, which is implemented using the Rust language and employs the BN128 (also called the alt BN256) elliptic curve for witness generation w. Powers of Tau (PoT) is universally Structured Reference String (SRS) used for the secure creation of polynomial commitments. All experiments are conducted in a Kubernetes environment that is deployed locally in a virtual machine setup with 32 GB RAM, Intel(R) Xeon(R) Silver 4216 Central Processing Unit (CPU) @ 2.10 GHz, 2 TB Solid State Drive (SSD) storage, and 10 MBsec bandwidth. The KZG commitment size for all types of health records remain constant, it depends on the encoding scheme, but for BN128, it is effectively around 64 bytes.

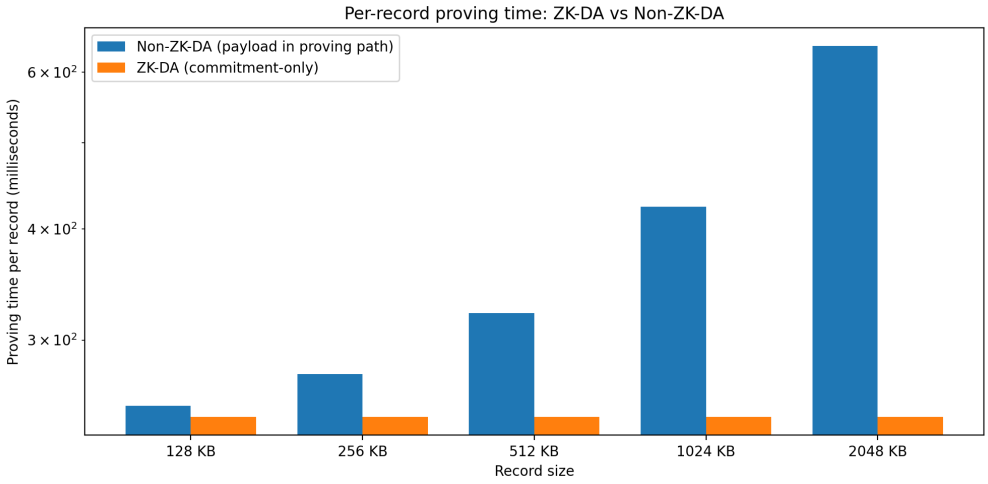

Figure (7.6) Proving Time for Varying Data Sizes (ZK-DA vs Non-ZK-DA)

7.8.1 Proving Time

From the scalability perspective, it is essential to know how much time it takes to generate the proof with and without the ZK-DA system. As discussed earlier, without ZK-DA, a Rollup system generated proofs on the raw data which ultimately increases the proof generating time (Proving Time) and also makes larger witness files (private input) to be used on the verification time. We started the evaluation to check the proving time with and without ZK-DA system. As shown in Figure 7.6, we observe the proving time for transactions with varying data size, we started with 128 KB EHR and images and ended with 2048 KB file size, where most of the files are images to create heavy payload. The proving time without ZK-DA increases with the record size. However, with the ZK-DA system, the proving time does not increase because it is only generating proofs over the commitments values generated with the KZG polynomial. With the PLONK ZKP system, the size of the generated proofs also remains constant (800 - 900) Bytes. Therefore, with this scheme, there is no need to worry about the proof sizes with the varying number of file size or batch sizes.

Varying the number of transactions with fixed record size (128, 256, 512) KB, proving the time differences between the ZK-DA and the non-ZK-DA system are shown in Figures 7.7, 7.8 and 7.9. As discussed earlier, for small size EHR (e.g., 128 KB) the proving time is lesser for ZK-DA system than for non-ZK-DA system, but not too significant. The reason is that the record size is the smallest in this case. Therefore, generating proof over raw data and generating proof over the commitment presents a more or less small difference. However, with the larger records sizes, the ZK-DA system starts to show its impact. We can observe in Figure 7.7, that we created batches for 2000 to 10000 transactions with fixed size healthcare records of 128 KB. The difference in proving time between two systems is not very large; however, the ZK-DA system shows optimistic results.

Figure 7.8 shows the difference in the proving time for the 256 KB record sizes. Here,

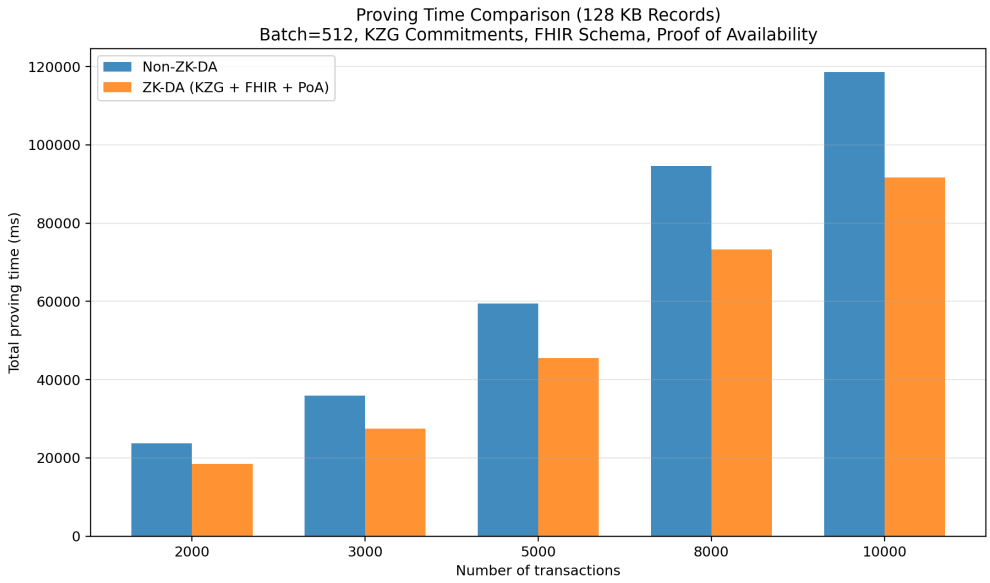

Figure (7.7) Proving Time for 128 KB Health Records with Varying Number of Transactions

we can see significant differences in proving time for the ZK-DA and non-ZK-DA systems. The file size increases and KZG polynomial commitments play a role in reducing the proving time significantly. For smaller file, ZK-DA system also have an overhead, that includes, KZG commitment generation, PoA opening, and FHIR schema checking. This overhead is influential for small records; therefore, it dominates both systems. ZK-DA significantly reduces the asymptotic cost of proving large records. However, as the record size increases, payload-dependent costs dominate the Non-ZK-DA proving time.

7.8.2 Speedup Factor

Generally, the speedup factor can be defined as:

$$Speedup = \frac{T_{non-ZK-DA}}{T_{ZK-DA}}$$

Figure 7.10 shows the speedup factor with a varying number of records. For small record sizes, the speedup factor is close to 1.0 because there is not much difference in the proving time for small records; however, with increasing file size, the speedup factor shows optimistic results. This speedup factor is calculated by the time taken by batching, generating commitments, sampling, and generating the final proof to submit on the blockchain. The time to validate the submitted proof on the permissioned blockchain is not included in the speedup factor calculation.

7.8.3 PoA Detection Probability

To measure the detection probability, we set some variables like u to present unavailable or corrupted data uploaded in the form of chunks on the DA layer and k is the sampling

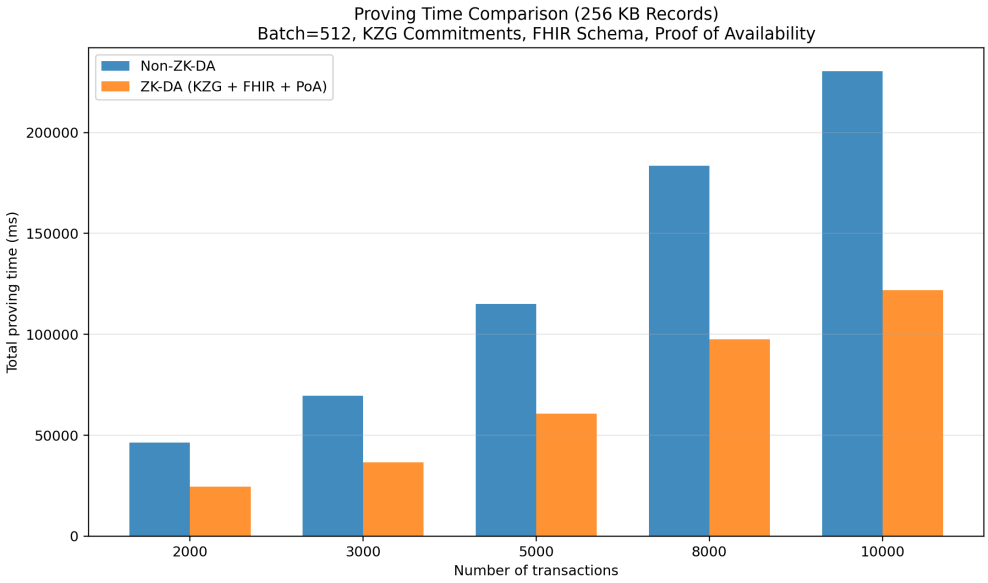

Figure (7.8) Proving Time for 256 KB Health Records with Varying Number of Transactions

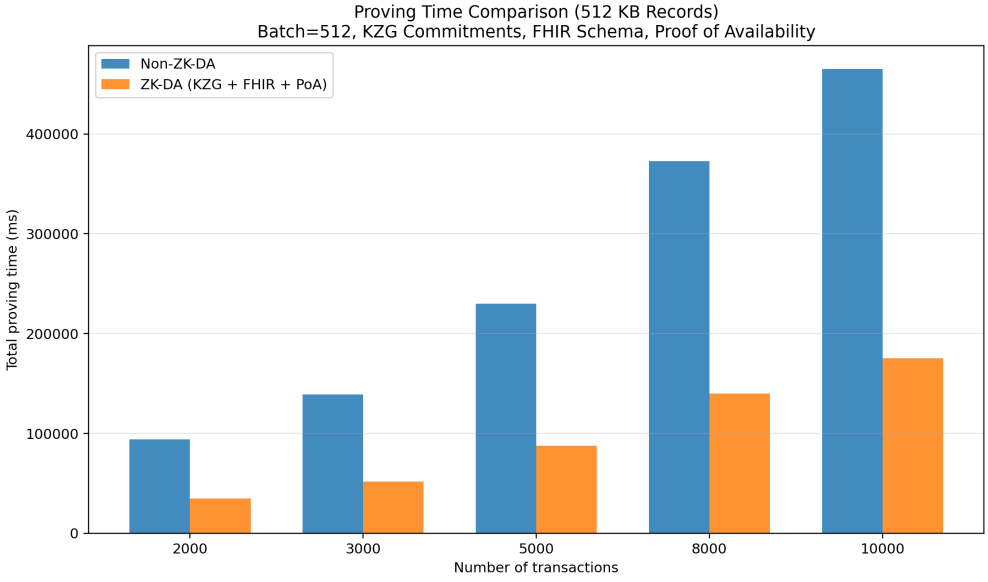

Figure (7.9) Proving Time for 512 KB Health Records with Varying Number of Transactions

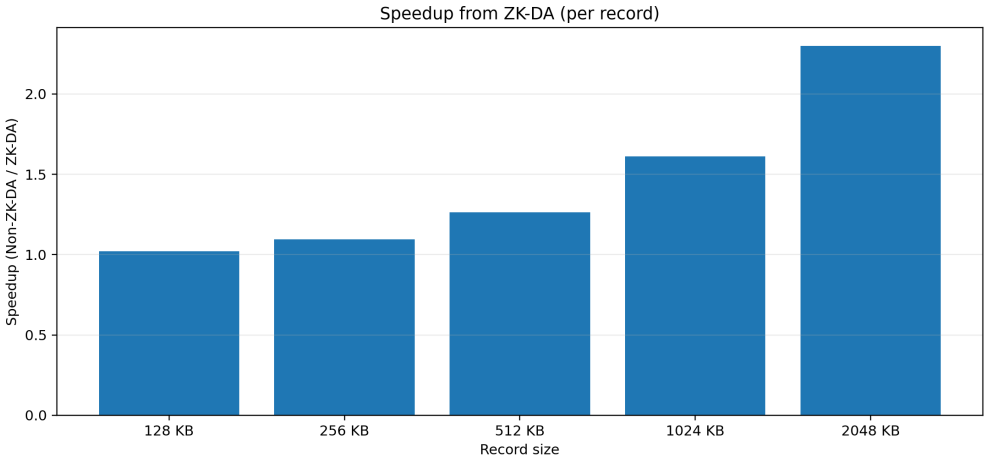

Figure (7.10) Speedup Factor with varying number of records

rate. We can also present these variables as:

$$P(detect)Detection - Probability = 1 - (1 - u)^k$$

where:

u = unavailable chunks/corrupted chunks

k = number of sampled chunks

This provides a security benefit for one round of PoA. Figure 7.11 presented the detection probability per PoA round compared to the varying sampling rate k. Here, PoA round means a round completed by the sampler with a set sampling rate k to detect unavailable data from the uploaded chunks on the DA layer. The detection probability in Figure 7.11 is presented with the number of uploaded chunks 128. Here, u is the percentage of unavailable or corrupted data uploaded to the DA layer in chunks. $u = 5\%$ means that 5% of the data is unavailable or corrupted in the uploaded chunks to check probability detection with a fixed sampling rate k. We can observe that with a lower percentage of data unavailability, the probability of detection is lower, as more unavailable data appears at the DA layer, the probability of detection increases. On the other side, by increasing the sampling rate to choose the random chunks, the probability of detection increases. As more random samples are selected to check the unavailable data, it helps to increase the detection probability.

In Figure 7.12 the same type of experiment is performed, but in this case the number of chunks uploaded to the DA layer is 256. It appears that by increasing the number of chunks in the DA layer, the probability of detection also increases. With value $u = 30\%$ the probability of detection is maximum, because the maximum percentage of data is not available on uploaded chunks. It shows that maximum nonavailability of the data can be detected with a lower sampling rate. However, by increasing the sampling rate to 30%, it can detect a small faction of unavailable or corrupted data.

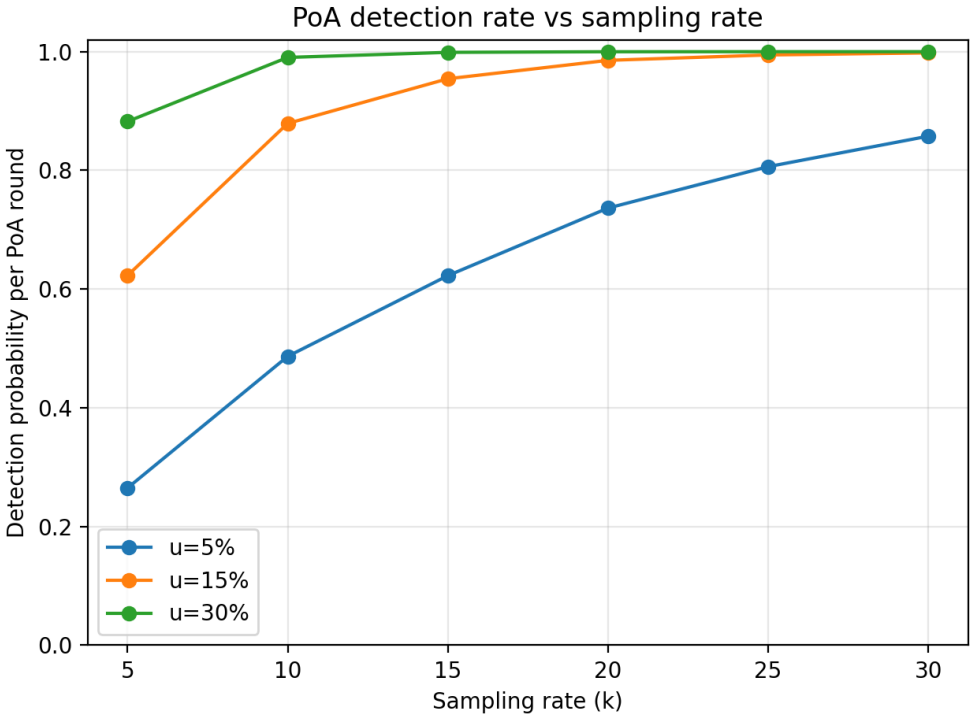

Figure (7.11) PoA Detection with Different Sampling Rate and No. of chunks = 128

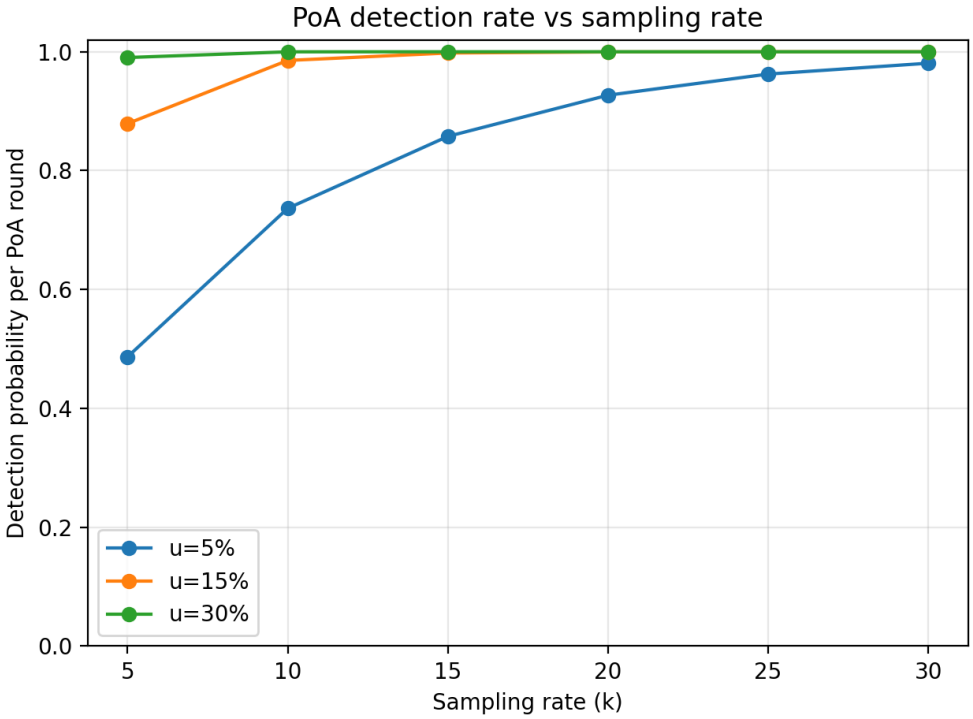

Figure (7.12) PoA Detection with Different Sampling Rate and No. of chunks = 256

7.8.4 Sampling Efficiency

In the concept of PoA, there is a trade-off of resources like bandwidth and computation for security. So, there is a cost to obtain this detection security. The sampling and publishing of chunks on the DA layer requires memory, computation, and bandwidth. If we say that, detection probability for one round of PoA:

$$P(detect)Detection - Probability = 1 - (1 - u)^k$$

Therefore, the PoA sampling efficiency per MB can be calculated as:

$$PoA - Efficiency_{MB} = \frac{P(detect)}{Bandwidth - per - round(MB)}$$

Bandwidth required for PoA round:

$$B_{MB} = \frac{k.ChunkSize}{1024}$$

Now:

$$PoA - Efficiency_{MB} = \frac{1 - (1 - u)^k}{\frac{k.ChunkSize}{1024}}$$

A question here is why it is meaningful, because bandwidth is an important factor in data availability systems; in this scenario, the data systems are bandwidth constrained. This calculation helps to find an optimum k sampling rate, where it provides maximum efficiency.

An intuition to calculate this efficiency is to find the probability of catching unavailable/corrupted data for every MB uploaded to the DA layer. Now we can see in Figure 7.13, we calculate the detection probability per MB against the sampling rate. To make it clear, on the Y-axis, this is not the probability and not a size, it is a rate. For example, if on the Y-axis, any value is 0.15 for a fixed sampling rate $k = 5$, then each sampled MB contributes 0.15 detection probability towards detecting unavailability. Therefore, it provides the amount of detection probability gained per MB of sampling cost. Higher sampling rate on the X-axis, more samples, and more MB downloaded. Higher detection probability gained per MB on the Y-axis, more efficient use of bandwidth, lower values, and more bandwidth consumed for detection. We can see in Figure 7.13, the low sampling rate, efficiency of the detection system is the maximum for a fixed value of u. As the sampling rate grows, the denominator factor grows, and the overall efficiency reduces. higher sampling rate, the detection probability saturates near 1, bandwidth consumption increases, and efficiency of the sampler decreases. These plots help us to understand the trade-off between bandwidth and detection probability. Different unavailability curves behave differently. For example, for high unavailability $u = 30\%$, detection is easy, few samples can detect data unavailability, so the efficiency of the system is high. This metric is important in the ZK-DA system because proofs ensure correctness, PoA ensures availability, and this availability cost is dominated by the bandwidth factor. It helps to

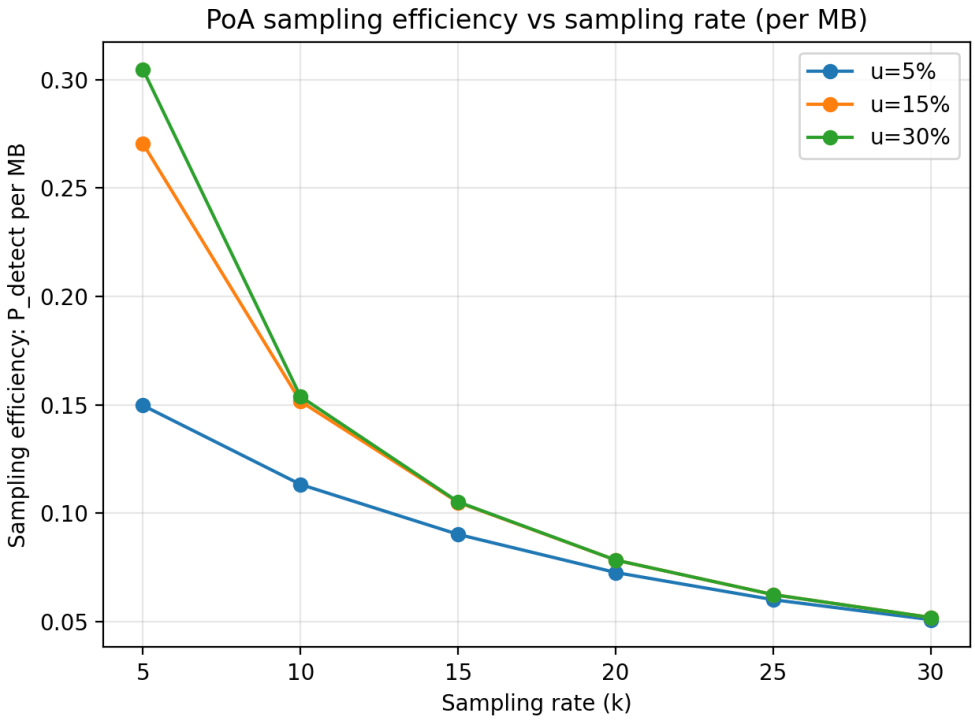

Figure (7.13) Sampling Efficiency: Detection Probability per MB

tune the sampling rate according to the requirements so as not to overspend bandwidth for more security.

In Figure 7.14, the PoA sampling efficiency for the detection probability per millisecond. this plot shows that for each fixed unavailability rate "u", how much detection power you get per millisecond of PoA time, when the sampler issues the requests in parallel.

If:

$$P(detect)Detection - Probability = 1 - (1-u)^k$$

parallel round latency with a round trip time (RTT):

$$L_{par}(k) = RTT + k.LCT_{perchunk}$$

Here, RTT is the round trip time and $LCT_{perchunk}$ is the Local check time per chunk. Therefore, a plot can be created with this formula:

$$PoA - Efficiency_{ms,par} = \frac{P(detect)}{L_{par}(k)} = \frac{1-(1-u)^k}{RTT.LCT_{perchunk}}$$

We can interpret this equation as; in one millisecond of PoA round trip time, what could be the probability to detect unavailable or corrupted data. On the X-axis, sampling rate k varies, but you can see that for the maximum sampling rate, the efficiency of the sampler is decreasing. It is more or less the same behavior as we discussed for the bandwidth case. For latency rate, the $P(detect)$ detection probability is lower, but as we

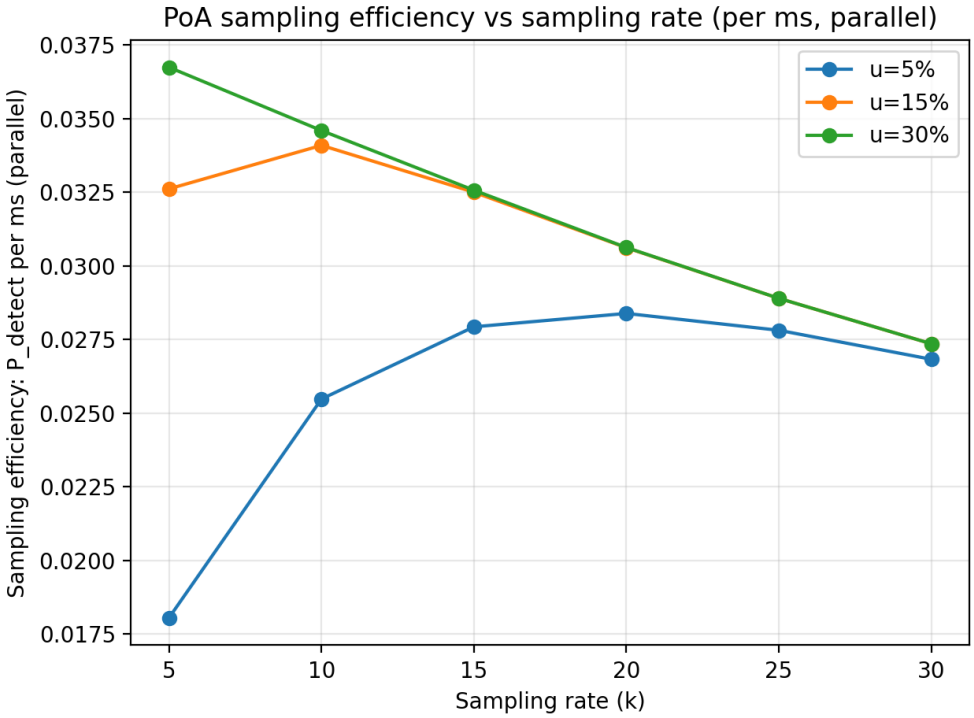

Figure (7.14) Sampling Efficiency: Detection Probability per millisecond

increase the sampling rate it increases very fast and also the latency increases a little. RTT dominates at first, so the efficiency increases initially for $u = 5$. However, at the higher sampling rate, $P(detect)$ saturates to 1, but we cannot exceed from 1, therefore the latency continues increase; therefore, the efficiency starts to decline.

$$PoA - Efficiency_{ms,par} == \frac{almost - 1}{RTT.LCT_{perchunk}}$$

At higher $u = 30\%$, green curve is higher, because most of the data is not available then detection becomes easier. It can be detected with a small sampling rate. $P(detect)$ becomes 1 at small k. The curve peaks earlier, high sampling rate is not effective.

For example, for a fixed sampling rate $k = 15$, a value on the Y-axis is 0.0270 with $u = 5\%$. It is telling , for one millisecond of PoA RTT, the efficiency rate is 0.0270, which is the probability mass to detect inaccessible or corrupted data. This value remains between 0.001 - 0.01, because probability is capped at 1.0 and latency is often tens of milliseconds.

7.9 Conclusion

In this chapter, we start our discussion by presenting a novel idea of ZK-DA, which provides a data privacy and scalability solution for healthcare data processed in the ZK-Rollup system. The general idea is to ensure the privacy of healthcare data by implementing the ZKP solution with the PLONK (ZK-SNARK) system and submitting a

single final proof on the blockchain system. We achieved scalability on the verification side (L-1 blockchain), but in the ZK-Rollup system, thousands of data access requests are managed, and ZKPs are generated for the verification of requested data without sharing the actual data. However, the scalability challenge lies in producing and managing a large number of proofs. We perform batching operations within the ZK-Rollup system to reduce the overhead on the underlying blockchain. A single proof submitted on the blockchain provides completeness and correctness for all batch transactions. However, generating these large numbers of proofs requires computational resources because in the ZK-Rollup systems these proofs are generated by reading the actual data, which leads to the longer proving time and larger circuit constraints to generate a proof. To overcome this issue, we introduce a novel concept of a DA (Data Availability) layer that provides the Proofs of Availability (PoA) of the data requested by the users. In this concept, the proofs are not generated by fetching the healthcare records from the database, but the commitment values are generated from these healthcare records, and these commitment values are provided as input to generate the proofs. These commitment values are generated by using data chunks in the KZG polynomial functions, which generate them with a constant size. In this way, the proof generator takes consistent time to create proofs for all types of transactions, whether there are EHR or imaging records. Subsequently, PoA are generated by performing a sampling operation to check the availability of the data. PoA is a probabilistic function that collects random chunks of data from the uploaded chunks on the DA layer, and each chunk collected is checked against its commitment value. If it is verified, then the data is correct and available; otherwise, that proof is rejected. In the results and evaluation section, we observe that this novel technique significantly reduces the proving time in a ZK-Rollup system. Provides strong scalability for large data records used to generate the proofs, as well as proofs of data availability and correctness by implementing the PoA concept. In addition, we also calculate the efficiency of the PoA system with varying sampling rates and data non-availability percentages. This allows us to offer recommendations for choosing the best sampling rate for both efficiency and detection rate.

Chapter 8

Conclusion and Future Work

This chapter provides an extensive summary of the overarching thesis and the research reported in earlier chapters. This chapter gives compelling reasons regarding how we manage security, privacy, and scalability issues in healthcare data, as well as innovative approaches to the issues highlighted. We shall also discuss the future course of action to further ideas in this thesis

8.1 Contribution Summary

- **Chapter 1.** This chapter started with a formal discussion of blockchain and distributed ledger technology. We discussed a major difference between blockchain and DLT in terms of blockchain objectives. We listed the objectives of this study and extensively addressed the research questions related to the issue. We discussed the thesis' structure in the previous section.

- **Chapter 2.** In the second chapter, we discussed the structures of public and private blockchain networks. How do hash functions manage transactions and what data are saved in blocks. Furthermore, we covered the blockchain levels and their functions. Because our thesis focused on the permissioned blockchain, it is necessary to examine its structure with a layered approach. How does it differ from the public blockchain. Moreover, we discussed Raft, which is a popular and fast consensus technique for permissioned blockchains, such as Hyperledger Fabric.

- **Chapter 3.** In the third chapter, we combined a sharded blockchain technique with healthcare data security, providing a promising research route for processing large amounts of healthcare data within a federated learning framework. The proposed method improves the resolution of various fundamental problems in centralized coordinator-based federated learning systems. Our strategy combines blockchain-based sharding technology with a federated learning network to train healthcare data. Where healthcare data protection and network scalability are important considerations. Individual peer nodes create localized machine learning models using data stored within healthcare devices. The locally developed models and parameters are shared with other peers via the blockchain platform.

- **Chapter 4.** In the fourth chapter, we focused our discussion on our core objective of solving scalability issue in blockchain. A scaling solution was discussed that integrates state channels with the Hyperledger Fabric blockchain. Diverse healthcare organizations used federated learning to train models on their different devices. The trained models are transmitted regularly over existing state channels to improve scalability and reduce blockchain consensus time and network bandwidth consumption. Finally, the results show that the use of state channels with blockchains provides significant savings in time and bandwidth.

- **Chapter 5.** The fifth chapter addresses the security and privacy issues related to healthcare data in the context of data sharing. The trained models are shared through a federated learning network, while the real data are kept private. However, hostile or faulty nodes can jeopardize data security and privacy, which are critical in the healthcare industry. We proposed an attribute-based access control (ABAC) system that is linked to blockchain to improve healthcare data access control. Furthermore, we added the trust attribute to NIST's established ABAC model to identify malicious nodes in the FL network and isolate them based on trust values calculated by aggregating the model quality values trained by these nodes, their behaviors, and their reputation values.

- **Chapter 6.** In the sixth chapter, we introduced zero-knowledge proofs (ZKPs) and discussed their types (interactive and non-interactive). We went on to explore two versions of non-interactive ZKP (ZK-SNARK and ZK-STARK), with an emphasis on the ZK-SNARK proof method. The ZK-SNARK proof system discusses Groth16 and Plonk, as well as their functional design. In this study, we generated and validated ZKPs using the PLONK proof system. This chapter also discusses the ZK-Rollup Layer-2 blockchain technique.

  Additionally, we evaluated and compared two layer-2 scaling strategies (ZK-Rollup and State Channels) in terms of latency, throughput, complexity, and management in the context of healthcare data security and privacy. The primary contribution is to highlight the significant distinctions in their roles and efficacy in the healthcare arena. We conducted this comparison analysis by simulating a healthcare data-sharing protocol. The comparative analysis section discusses the benefits and drawbacks of each strategy.

- **Chapter 7.** In the seventh chapter, we begun our discussion by introducing a unique ZK-DA concept that provides a data privacy and scalability solution for healthcare data handled in the ZK-Rollup platform. The overall goal is to protect the privacy of healthcare data by integrating the ZKP solution with the PLONK (ZK-SNARK) system to submit a single final proof to the blockchain. We achieved scalability on the verification side (L-1 blockchain), but in the ZK-Rollup system, thousands of data access requests are handled, and ZKPs are created to verify requested data without disclosing the actual data. We use batching operations in

the ZK-Rollup system to decrease the overhead on the underlying blockchain. A single proof filed on the blockchain ensures that all batch transactions are complete and accurate. However, creating these vast numbers of proofs demands computing resources since in ZK-Rollup systems, proofs are formed by reading actual data, resulting in a longer proving time and bigger circuit restrictions to construct a proof. To address this issue, we presented a unique idea, called a DA (Data Availability) layer, which delivers Proofs of Availability (PoA) for the data requested by users. In this notion, proofs are not formed by retrieving healthcare information from a database; rather, commitment values are generated from these healthcare records and used as input to generate proofs. These commitment values are constructed using data chunks from KZG polynomial functions, which are fixed in size. In this approach, the proof generator takes constant time to generate proofs for all sorts of transaction, regardless of whether they use EHR or imaging information. Subsequently, PoA are formed by undertaking a sampling process to ensure data availability. PoA is a probabilistic function that takes random chunks of data from uploaded chunks on the DA layer and compares them to their commitment values. If validated, the data are accurate and available; otherwise, the evidence is denied. In the findings and assessment section, we can see that this unique method greatly decreases the proving time in a ZK-Rollup system. Implementing the PoA idea ensures excellent scalability for huge data records required to construct proofs, as well as evidence of data availability and accuracy. In addition, we calculate the PoA system's efficiency for different sampling rates and data non-availability percentages. This allows us to make suggestions on the ideal sample rate for efficiency and detection rate.

8.2 Future Work

There is potential for further research into topics discussed in the thesis.

8.2.1 Adaptive Data Availability Approach

In future work, an adaptive sampling approach can be explored based on data criticality, risk factor, and user type can be explored. It can also open doors for context-aware sampling, where abnormal or highly critical data records trigger higher sampling rates. Healthcare data can be split into layers (EHR records, imaging, and lab results) with different sampling rates. This approach can raise the challenges of sampling policy creation, and adaptive sampling does not weaken detection probability.

8.2.2 Cross-chain Healthcare Interoperability

The cross-chain ZK-Rollup system facilitates interoperability among national or regional healthcare systems. This approach enables the achievement of optimal privacy without the need of transmitting raw data. It can also assist in addressing the double expenditure

issue. ZK-bridges facilitate the verification of medical claims and settlements across healthcare organizations. This method may provide challenges with standardized proof creation and DA guarantees across systems. Robust rules are necessary for cross-chain ZK-Rollup systems.

8.2.3 Trusted Execution Environment (TEE) with ZKP

A hybrid architecture may be established by integrating Zero-Knowledge Proofs (ZKP) with Trusted Execution Environments (TEE) to expedite the verification of extensive medical data sets. This environment can also provide safe pre-processing of photos or extensive genetic data. This hybrid system can improve the throughput of PoA samplers. This hybrid method may also provide challenges when ensuring security against side-channel attacks and in integrating Trusted Execution Environments (TEEs) with Zero-Knowledge Proof (ZKP) systems.

8.2.4 Complex ZK Query Expressiveness

In future work, ZK circuits are created that can handle complex medical queries or temporal queries (e.g., "Hemoglobin increased in last 6 months") or conditional queries (e.g., "If patient is diabetic, verify insulin dosage"). For such complex ZKP circuits, Zero Knowledge Virtual Machine (ZKVM) can be integrated with the ZK-Rollup systems. A ZKVM provides a solid platform to handle such complex queries, where high level languages or domain-specific languages (DSLs) are used to create the circuits. For complex ZKP circuits, circuit complexity explosion is a major challenge for expressive medical logic. It also raises challenges for verifier efficiency because complex queries take more time to generate a proof.

Bibliography

[1] Abdullatif Albaseer, Bekir Sait Ciftler, Mohamed Abdallah, and Ala Al-Fuqaha. Exploiting unlabeled data in smart cities using federated edge learning. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1666–1671. IEEE, 2020.

[2] Gaspard Anthoine, Jean-Guillaume Dumas, Mélanie de Jonghe, Aude Maignan, Clément Pernet, Michael Hanling, and Daniel S Roche. Dynamic proofs of retrievability with low server storage. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 537–554, 2021.

[3] Giuseppe Ateniese, Long Chen, Mohammad Etemad, and Qiang Tang. Proof of storage-time: Efficiently checking continuous data availability. *Cryptology ePrint Archive*, 2020.

[4] Leo Maxim Bach, Branko Mihaljevic, and Mario Zagar. Comparative analysis of blockchain consensus algorithms. In *2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 1545–1550. Ieee, 2018.

[5] Gioele Bigini, Valerio Freschi, and Emanuele Lattanzi. A review on blockchain for the internet of medical things: Definitions, challenges, applications, and vision. *Future Internet*, 12(12):208, 2020.

[6] Gioele Bigini, Valerio Freschi, and Emanuele Lattanzi. A review on blockchain for the internet of medical things: Definitions, challenges, applications, and vision. *Future Internet*, 12(12):208, 2020.

[7] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388, 2019.

[8] Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Efficient polynomial commitment schemes for multiple points and polynomials. *Cryptology ePrint Archive*, 2020.

[9] Vitalik Buterin. The dawn of hybrid layer 2 protocols. *Vitalik Buterin's website*, 2019.

[10] Stefanos Chaliasos, Itamar Reif, Adrià Torralba-Agell, Jens Ernstberger, Assimakis Kattis, and Benjamin Livshits. Analyzing and benchmarking zk-rollups. In *6th Conference on Advances in Financial Technologies (AFT 2024)*, pages 6–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

[11] Zeng Chen, Weidong Xu, Bingtao Wang, and Hua Yu. A blockchain-based preserving and sharing system for medical data privacy. *Future Generation Computer Systems*, 124:338–350, 2021.

[12] Shumo Chu and Sophia Wang. The curses of blockchain decentralization. *arXiv preprint arXiv:1810.02937*, 2018.

[13] Kristtopher K Coelho, Michele Nogueira, Alex B Vieira, Edelberto F Silva, and José Augusto M Nacif. A survey on federated learning for security and privacy in healthcare applications. *Computer Communications*, 207:113–127, 2023.

[14] Kristtopher K Coelho, Michele Nogueira, Alex B Vieira, Edelberto F Silva, and José Augusto M Nacif. A survey on federated learning for security and privacy in healthcare applications. *Computer Communications*, 207:113–127, 2023.

[15] Saptarshi Das, Barsha Mitra, Vijayalakshmi Atluri, Jaideep Vaidya, and Shamik Sural. Policy engineering in rbac and abac. In *From Database to Cyber Security: Essays Dedicated to Sushil Jajodia on the Occasion of His 70th Birthday*, pages 24–54. Springer, 2018.

[16] Joshua D Drake and John C Worsley. *Practical PostgreSQL*. " O'Reilly Media, Inc.", 2002.

[17] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[18] Hyperledger Fabric. Deploying a smart contract on a channel, 2024. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.5/deploy_chaincode.html.

[19] Muhammad Ashraf Fauzi, Norazha Paiman, and Zarina Othman. Bitcoin and cryptocurrency: Challenges, opportunities and future works. *The Journal of Asian Finance, Economics and Business*, 7(8):695–704, 2020.

[20] Dankrad Feist. New sharding design with tight beacon and shard block integration, 2022.

[21] Libo Feng, Hui Zhang, Wei-Tek Tsai, and Simeng Sun. System architecture for high-performance permissioned blockchains. *Frontiers of Computer Science*, 13(6):1151–1165, 2019.

[22] Anmin Fu, Xianglong Zhang, Naixue Xiong, Yansong Gao, Huaqun Wang, and Jing Zhang. Vfl: A verifiable federated learning with privacy-preserving for big data in industrial iot. *IEEE Transactions on Industrial Informatics*, 18(5):3316–3326, 2020.

[23] Mayra Rosario Fuentes. Cybercrime and other threats faced by the healthcare industry, 2017.

[24] Weichao Gao, William G Hatcher, and Wei Yu. A survey of blockchain: Techniques, applications, and challenges. In *2018 27th international conference on computer communication and networks (ICCCN)*, pages 1–11. IEEE, 2018.

[25] Vincent Gramoli, Len Bass, Alan Fekete, and Daniel W Sun. Rollup: Non-disruptive rolling upgrade with fast consensus-based dynamic reconfigurations. *IEEE Transactions on Parallel and Distributed Systems*, 27(9):2711–2724, 2015.

[26] Suyash Gupta, Sajjad Rahnama, and Mohammad Sadoghi. Permissioned blockchain through the looking glass: Architectural and implementation lessons learned. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 754–764. IEEE, 2020.

[27] Russ Housley. Public key infrastructure (pki). *The internet encyclopedia*, 2004.

[28] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, et al. Guide to attribute based access control (abac) definition and considerations (draft). *NIST special publication*, 800(162):1–54, 2013.

[29] Chengpeng Huang, Rui Song, Shang Gao, Yu Guo, and Bin Xiao. Data availability and decentralization: New techniques for zk-rollups in layer 2 blockchain networks. *arXiv preprint arXiv:2403.10828*, 2024.

[30] Dongyan Huang, Xiaoli Ma, and Shengli Zhang. Performance analysis of the raft consensus algorithm for private blockchains. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(1):172–181, 2019.

[31] Ioana Paula Iacoban. Statefl: State channels powered federated learning. *Master's Thesis*, 2024.

[32] Uzma Jafar, Mohd Juzaiddin Ab Aziz, and Zarina Shukur. Blockchain for electronic voting system—review and open research challenges. *Sensors*, 21(17):5874, 2021.

[33] The HIPAA Journal. Healthcare data breach statistics, 2024. URL: https://www.hipaajournal.com/healthcare-data-breach-statistics/.

[34] Ghassan Karame. On the security and scalability of bitcoin's blockchain. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1861–1862, 2016.

[35] Aniket Kate, Gregory M Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In *International conference on the theory and application of cryptology and information security*, pages 177–194. Springer, 2010.

[36] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[37] Ben Laurie and Richard Clayton. Proof-of-work proves not to work; version 0.2. In *Workshop on economics and information, security*, 2004.

[38] Wenyu Li, Chenglin Feng, Lei Zhang, Hao Xu, Bin Cao, and Muhammad Ali Imran. A scalable multi-layer pbft consensus for blockchain. *IEEE Transactions on Parallel and Distributed Systems*, 32(5):1146–1160, 2020.

[39] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Transactions on Industrial Informatics*, 16(3):2134–2143, 2019.

[40] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Transactions on Vehicular Technology*, 69(4):4298–4311, 2020.

[41] Jason Luu and Edward J Imwinkelried. The challenge of bitcoin pseudo-anonymity to computer forensics. *Criminal Law Bulletin*, 52(1):191–236, 2016.

[42] Shengchen Ma and Xing Zhang. Integrating blockchain and zk-rollup for efficient healthcare data privacy protection system via ipfs. *Scientific Reports*, 14(1):11746, 2024.

[43] Soubhagya Ranjan Mallick, Rakesh Kumar Lenka, Pradyumna Kumar Tripathy, D Chandrasekhar Rao, Suraj Sharma, and Niranjan Kumar Ray. A lightweight, secure, and scalable blockchain-fog-iomt healthcare framework with ipfs data storage for healthcare 4.0. *SN Computer Science*, 5(1):198, 2024.

[44] André Henrique Mayer, Cristiano André da Costa, and Rodrigo da Rosa Righi. Electronic health records in a blockchain: A systematic review. *Health informatics journal*, 26(2):1273–1288, 2020.

[45] André Henrique Mayer, Cristiano André da Costa, and Rodrigo da Rosa Righi. Electronic health records in a blockchain: A systematic review. *Health informatics journal*, 26(2):1273–1288, 2020.

[46] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. A review on consensus algorithm of blockchain. In *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 2567–2572. Ieee, 2017.

[47] Anshul Mittal and Swati Aggarwal. Hyperparameter optimization using sustainable proof of work in blockchain. *Frontiers in Blockchain*, 3:23, 2020.

[48] Ervin Moore, Ahmed Imteaj, Shabnam Rezapour, and M. Hadi Amini. A survey on secure and private federated learning using blockchain: Theory and application in resource-constrained computing. *IEEE Internet of Things Journal*, 10(24):21942–21958, 2023. doi:10.1109/JIOT.2023.3313055.

[49] Ervin Moore, Ahmed Imteaj, Shabnam Rezapour, and M. Hadi Amini. A survey on secure and private federated learning using blockchain: Theory and application in resource-constrained computing. *IEEE Internet of Things Journal*, 10(24):21942–21958, 2023. doi:10.1109/JIOT.2023.3313055.

[50] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[51] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[52] Raushan Myrzashova, Saeed Hamood Alsamhi, Alexey V Shvetsov, Ammar Hawbani, and Xi Wei. Blockchain meets federated learning in healthcare: A systematic review with challenges and opportunities. *IEEE Internet of Things Journal*, 2023.

[53] Raushan Myrzashova, Saeed Hamood Alsamhi, Alexey V Shvetsov, Ammar Hawbani, and Xi Wei. Blockchain meets federated learning in healthcare: A systematic review with challenges and opportunities. *IEEE Internet of Things Journal*, 2023.

[54] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Available at SSRN 3440802*, 2008.

[55] Lydia D Negka and Georgios P Spathoulas. Blockchain state channels: A state of the art. *IEEE Access*, 9:160277–160298, 2021.

[56] Trinh Nguyen-Phan, Victoria Lemieux, and Tung Bui. A decentralized information governance framework for implementing blockchain in healthcare: An exploratory analysis. 2025.

[57] Emmanuel Nyaletey, Reza M Parizi, Qi Zhang, and Kim-Kwang Raymond Choo. Blockipfs-blockchain-enabled interplanetary file system for forensic and trusted data traceability. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 18–25. IEEE, 2019.

[58] Aditya Pathak, Irfan Al-Anbagi, and Howard J Hamilton. Tabi: Trust-based abac mechanism for edge-iot using blockchain technology. *IEEE Access*, 11:36379–36398, 2023.

[59] Shiva Raj Pokhrel and Jinho Choi. Improving tcp performance over wifi for internet of vehicles: A federated learning approach. *IEEE transactions on vehicular technology*, 69(6):6798–6802, 2020.

[60] Darshika Ravindra Pongallu, Shreyas Setlur Arun, and Rohit Verma. Securemedzk-a blockchain-based approach with zero-knowledge rollups for secure ehr. In *2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC)*, pages 1–9. IEEE, 2025.

[61] Attia Qammar, Ahmad Karim, Huansheng Ning, and Jianguo Ding. Securing federated learning with blockchain: a systematic literature review. *Artificial Intelligence Review*, 56(5):3951–3985, 2023.

[62] Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 198–214. IEEE, 2022.

[63] Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 198–214. IEEE, 2022.

[64] Xidi Qu, Shengling Wang, Qin Hu, and Xiuzhen Cheng. Proof of federated learning: A novel energy-recycling consensus algorithm. *IEEE Transactions on Parallel and Distributed Systems*, 32(8):2074–2085, 2021.

[65] Neal Quinn. Cyber attacks on healthcare soar: What lies ahead in 2024, 2024. URL: https://www.healthcarebusinesstoday.com/cyber-attacks-on-healthcare-soar-what-lies-ahead-in-2024/.

[66] Gabriel Antonio F Rebello, Gustavo F Camilo, Lucas Airam C de Souza, Maria Potop-Butucaru, Marcelo Dias de Amorim, Miguel Elias M Campista, and Luís Henrique MK Costa. A survey on blockchain scalability: From hardware to layer-two protocols. *IEEE Communications Surveys & Tutorials*, 26(4):2411–2458, 2024.

[67] Research and Markets. Electronic medical records - market share analysis, industry trends and statistics, growth forecasts 2021 - 2029, 2024. URL: https://www.researchandmarkets.com/reports/4520273/electronic-medical-records-market-share.

[68] Mohammad Ali Saberi, Mehdi Adda, and Hamid Mcheick. Break-glass conceptual model for distributed ehr management system based on blockchain, ipfs and abac. *Procedia Computer Science*, 198:185–192, 2022.

[69] SafetyDetectives. Healthcare cybersecurity: The biggest stats and trends in 2024, 2024. URL: https://www.safetydetectives.com/blog/healthcare-cybersecurity-statistics/.

[70] Cyprian Omukhwaya Sakwa, Andrew Omala Anyembe, and Fagen Li. A survey of folding-based zero-knowledge proofs. *Information Sciences*, 724:122698, 2026. URL: https://www.sciencedirect.com/science/article/pii/S002002552500831X, doi:10.1016/j.ins.2025.122698.

[71] Rishi Saripalle, Christopher Runyan, and Mitchell Russell. Using hl7 fhir to achieve interoperability in patient health record. *Journal of biomedical informatics*, 94:103188, 2019.

[72] Cosimo Sguanci, Roberto Spatafora, and Andrea Mario Vergani. Layer 2 blockchain scaling: A survey. *arXiv preprint arXiv:2107.10881*, 2021.

[73] Jahan Zeb Shahid, Stelvio Cimato, and Zia Muhammad. A sharded blockchain architecture for healthcare data. In *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1794–1799, 2024. doi:10.1109/COMPSAC61105.2024.00283.

[74] Jahanzeb Shahid and Stelvio Cimato. A state channel based approach to address scalability of healthcare data sharing. In *2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1861–1866, 2025. doi:10.1109/COMPSAC65507.2025.00255.

[75] Elham A Shammar, Ammar T Zahary, and Asma A Al-Shargabi. An attribute-based access control model for internet of things using hyperledger fabric blockchain. *Wireless Communications and Mobile Computing*, 2022(1):6926408, 2022.

[76] Chenhao Shi, Hao Chen, Ruibang Liu, and Guoqiang Li. Data-flow-based normalization generation algorithm of r1cs for zero-knowledge proof. In *2023 IEEE 28th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 191–197, 2023. doi:10.1109/PRDC59308.2023.00031.

[77] Moirangthem Biken Singh, Himanshu Singh, and Ajay Pratap. Energy-efficient and privacy-preserving blockchain based federated learning for smart healthcare system. *IEEE Transactions on Services Computing*, 2023.

[78] Moirangthem Biken Singh, Himanshu Singh, and Ajay Pratap. Energy-efficient and privacy-preserving blockchain based federated learning for smart healthcare system. *IEEE Transactions on Services Computing*, 2023.

[79] Moirangthem Biken Singh, Himanshu Singh, and Ajay Pratap. Energy-efficient and privacy-preserving blockchain based federated learning for smart healthcare system. *IEEE Transactions on Services Computing*, 2023.

[80] Bikramaditya Singhal, Gautam Dhameja, and Priyansu Sekhar Panda. How blockchain works. In *Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions*, pages 31–148. Springer, 2018.

[81] Tianshu Song, Yongxin Tong, and Shuyue Wei. Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2577–2586. IEEE, 2019.

[82] Shravan Srinivasan, Ioanna Karantaidou, Foteini Baldimtsi, and Charalampos Papamanthou. Batching, aggregation, and zero-knowledge proofs in bilinear accumulators. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2719–2733, 2022.

[83] Xiaoqiang Sun, F. Richard Yu, Peng Zhang, Zhiwei Sun, Weixin Xie, and Xiang Peng. A survey on zero-knowledge proof in blockchain. *IEEE Network*, 35(4):198–205, 2021. doi:10.1109/MNET.011.2000473.

[84] Ali Sunyaev. Distributed ledger technology. In *internet Computing*, pages 265–299. Springer, 2020.

[85] Isti Surjandari, Harman Yusuf, Enrico Laoh, and Rayi Maulida. Designing a permissioned blockchain network for the halal industry using hyperledger fabric with multiple channels and the raft consensus mechanism. *Journal of Big Data*, 8(1):10, 2021.

[86] Hamed Taherdoost. Blockchain technology and artificial intelligence together: a critical review on applications. *Applied Sciences*, 12(24):12948, 2022.

[87] Dejan Vujičić, Dijana Jagodić, and Siniša Ranđić. Blockchain technology, bitcoin, and ethereum: A brief overview. In *2018 17th international symposium infoteh-jahorina (infoteh)*, pages 1–6. IEEE, 2018.

[88] Taochun Wang, Qingshan Wu, Jian Chen, Fulong Chen, Dong Xie, and Huimin Shen. Health data security sharing method based on hybrid blockchain. *Future Generation Computer Systems*, 153:251–261, 2024. URL: https://www.sciencedirect.com/science/article/pii/S0167739X23004478, doi:10.1016/j.future.2023.11.032.

[89] Yajing Wang, Jingsha He, Nafei Zhu, Yuzi Yi, Qingqing Zhang, Hongyu Song, and Ruixin Xue. Security enhancement technologies for smart contracts in the blockchain: A survey. *Transactions on Emerging Telecommunications Technologies*, 32(12):e4341, 2021.

[90] Yajing Wang, Jingsha He, Nafei Zhu, Yuzi Yi, Qingqing Zhang, Hongyu Song, and Ruixin Xue. Security enhancement technologies for smart contracts in the blockchain: A survey. *Transactions on Emerging Telecommunications Technologies*, 32(12):e4341, 2021.

[91] Yang Xiao, Ning Zhang, Wenjing Lou, and Y Thomas Hou. A survey of distributed consensus protocols for blockchain networks. *IEEE communications surveys & tutorials*, 22(2):1432–1465, 2020.

[92] Xincheng Yan, Jianhua Wu, Na Zhou, Zhihong Jiang, Juqin Wu, Jianhui Yin, and Ying Liu. Blockchain-based access control model for security attributes in the internet of things. In *2023 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, pages 95–101. IEEE, 2023.

[93] Qinglin Yang, Huawei Huang, Zhaokang Yin, Yue Lin, Qinde Chen, Xiaofei Luo, Taotao Li, Xiulong Liu, and Zibin Zheng. The state-of-the-art and promising future of blockchain sharding. *IEEE Communications Magazine*, 2024.

[94] Ibrar Yaqoob, Khaled Salah, Raja Jayaraman, and Yousof Al-Hammadi. Blockchain for healthcare data management: opportunities, challenges, and future recommendations. *Neural Computing and Applications*, 34(14):11475–11490, 2022.

[95] Abbas Yazdinejad, Ali Dehghantanha, Hadis Karimipour, Gautam Srivastava, and Reza M Parizi. A robust privacy-preserving federated learning model against model poisoning attacks. *IEEE Transactions on Information Forensics and Security*, 2024.

[96] Abbas Yazdinejad, Ali Dehghantanha, Hadis Karimipour, Gautam Srivastava, and Reza M Parizi. A robust privacy-preserving federated learning model against model poisoning attacks. *IEEE Transactions on Information Forensics and Security*, 2024.

[97] Yufeng Zhan, Jie Zhang, Zicong Hong, Leijie Wu, Peng Li, and Song Guo. A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing*, 10(2):1035–1044, 2021.

[98] Fan Zhang, Shaoyong Guo, Xuesong Qiu, Siya Xu, Feng Qi, and Zhili Wang. Federated learning meets blockchain: State channel-based distributed data-sharing trust supervision mechanism. *IEEE Internet of Things Journal*, 10(14):12066–12076, 2021.

[99] Xu Zheng and Zhipeng Cai. Privacy-preserved data sharing towards multiple parties in industrial iots. *IEEE Journal on Selected Areas in Communications*, 38(5):968–979, 2020.

[100] Chen Zhonghua, SB Goyal, and Anand Singh Rajawat. Smart contracts attribute-based access control model for security & privacy of iot system using blockchain and edge computing. *The Journal of Supercomputing*, 80(2):1396–1425, 2024.

[101] Chen Zhonghua, SB Goyal, and Anand Singh Rajawat. Smart contracts attribute-based access control model for security & privacy of iot system using blockchain and edge computing. *The Journal of Supercomputing*, 80(2):1396–1425, 2024.

[102] Liyi Zhou, Kaihua Qin, Antoine Cully, Benjamin Livshits, and Arthur Gervais. On the just-in-time discovery of profit-generating transactions in defi protocols. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 919–936. IEEE, 2021.

[103] Saide Zhu, Zhipeng Cai, Huafu Hu, Yingshu Li, and Wei Li. zkcrowd: a hybrid blockchain-based crowdsourcing platform. *IEEE Transactions on Industrial Informatics*, 16(6):4196–4205, 2019.

[104] Valentin Zieglmeier, Gabriel Loyola Daiqui, and Alexander Pretschner. Decentralized inverse transparency with blockchain. *Distributed Ledger Technologies: Research and Practice*, 2(3):1–28, 2023.

Appendix A

Content for ZK-DA

A.1 Formal Definitions

A.1.1 ZKP

A Zero-Knowledge Proof (ZKP) is a cryptographic method allowing one party (the prover) to prove to another (the verifier) that a statement is true without revealing any, or very little, of the underlying secret data.

A.1.2 ZK-Rollup

A ZK (Zero-Knowledge) Rollup is a Layer 2 scaling solution that increases blockchain throughput (like Ethereum) by processing transactions off-chain and bundling them into a single batch, significantly reducing fees. It uses cryptographic "validity proofs" to submit a cryptographic proof to the main chain, guaranteeing transaction correctness without disclosing private details.

A.1.3 Data Availability

Data availability ensures that all data required to verify or reconstruct a system's state is publicly retrievable within a bounded time, even in the presence of adversarial actors.

A.1.4 ZK-DA

Data availability ensures that all data required to verify or reconstruct a system's state is publicly retrievable within a bounded time, even in the presence of adversarial actors.

A.1.5 Proof of Availability

PoA provides a probabilistic guarantee that data claimed to be available is indeed retrievable, without requiring full data download.

A.1.6 PLONK

PLONK ("Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge") is an efficient, universal, and non-interactive zero-knowledge proof (ZKP) system. It enables one party (prover) to prove the correctness of a statement without revealing sensitive information, using a single "trusted setup" for any circuit, making it highly flexible for blockchain scaling and privacy.

A.2 Mathematical Notations

G_1, G_2 are cyclic groups

SRS = Structured Reference String:) it is a set of public, trusted parameters derived from hidden, random values (toxic waste) used by provers and verifiers to create and check proofs.

τ = The Powers of Tau is a secure, multi-party computation (MPC) ceremony used to generate universal, public parameters (structured reference string) for zk-SNARKs.

C = It is the output of KZG polynomial function, a commitment value

π = A single proof generated by the PLONK function

w = Witness Value = n Zero-Knowledge Proofs (ZKP), the witness value (often denoted as w or private input) is the secret information known only to the prover, which is used to demonstrate that a specific statement is true without revealing the secret itself.

Appendix B

Content for Trust Management Module

B.1 Formal Definitions

B.1.1 Federated Learning

Federated learning is a decentralized machine learning approach that trains AI models across multiple devices or servers (clients) holding local data, without exchanging the actual data. Instead of uploading data to a central server, devices train a local model and send only model updates (e.g., weights or gradients) to be aggregated, enhancing privacy and reducing data transmission.

B.1.2 Attribute-based Access Control

Attribute-Based Access Control (ABAC) is an advanced, dynamic authorization model that grants or denies access to resources by evaluating specific, real-time characteristics—or attributes—of the user, resource, action, and environment, rather than relying solely on static roles. It provides fine-grained, policy-based security (e.g., "Allow Engineers to edit files if it's during business hours and they are in the office").

Interplanetary File System

The InterPlanetary File System (IPFS) is a decentralized, peer-to-peer (P2P) protocol for storing, sharing, and accessing files, websites, and data. Unlike traditional HTTP, which locates data on a specific server, IPFS uses content-addressing to retrieve data from a network of nodes, enhancing security, reliability, and censorship resistance.

B.1.3 Homomorphic Encryption

Homomorphic encryption is an advanced, privacy-preserving cryptographic technique that allows computations (such as addition or multiplication) to be performed directly on encrypted data (ciphertext) without decrypting it first.

B.1.4 Differential Privacy

Differential privacy (DP) is a rigorous mathematical framework that enables the analysis of datasets to extract aggregate patterns while ensuring individual user data remains private

B.1.5 Shapley Value Algorithm

The Shapley value algorithm is a method from cooperative game theory used to fairly distribute the total gain or cost of a "game" among its participants (players) based on their average marginal contribution across all possible coalitions.

B.1.6 NIST ABAC Model

The NIST Attribute Based Access Control (ABAC) model, defined in SP 800-162, is a flexible, dynamic authorization framework that grants access to resources based on attributes—subject (user), object (resource), action, and environment—rather than just roles or identities.

B.2 Mathematical Notations

q_i = model's quality of a node i
k = it is the iteration number running
q_i^k = model's quality of a node i at $k-th$ iteration
b_i = behavior of node a node i
R_i = reputation of a node i
T_i^k = model update time of a node i at $k-th$ iteration
δ_i^k = normalized delay of node i at $k-th$ iteration
N_i =node i participating in the FL task
b_i^k = behavior of node i at $k-th$ iteration
NQ_i = list of model's quality values for all participating nodes
D_i^k = delay factor of node i at k-th iteration
η_i^k = learning rate of node i at k-th iteration
M = total number of nodes participating in a FL task
T = total number of iterations to complete one FL task
BR_i = list of behavior values for all participating nodes
$Trust_i$ = Trust value of a node i
ϕ_i = Calculated shapley value of any node i, participating in FL task
R_i = Reputation value of any node i, calculated by the shapely value aglorithm

La borsa di dottorato cofinanziata con risorse dell'Unione europea-NextGeneration EU Piano Nazionale di Ripresa e Resilienza Missione 4 – Componente 1 – Riforma 4.1 Riforma dei Dottorati - Inv. 4.1 Borse Generici Ricerca PNRR – CUP J11J22002120006