DESIGN AND DEVELOPMENT OF SECURITY SOLUTIONS IN BLOCKCHAIN

Thesis submitted to

Delhi Technological University

for the award of the degree of

DOCTOR OF PHILOSOPHY

in

Information Technology

by

Anita Thakur

(2K21/PHDIT/504)

Under the Supervision of

Dr. Virender Ranga

&

Dr. Ritu Agarwal

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly known as Delhi College of Engineering)
NEW DELHI-110042, INDIA

(MAY, 2025)

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

Candidate's Declaration

I hereby declare that the work being presented in this thesis entitled "**Design and Development of Security Solutions in Blockchain**" in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** and submitted in the Department of Information Technolog, Delhi Technological University, Delhi, is an authentic record of my own work carried out during a period from January 2022 to May 2025 under the supervision of **Dr.Virender Ranga**, Associate Professor, Department of Information Technology, Delhi Technological University, Delhi and **Dr.Ritu Agarwal**, Associate Professor, Department of Information Technology, Delhi Technological University, Delhi.

I have not submitted the matter presented in this thesis for the award of any other degree at this or other Institute.

(**Anita Thakur**)

(Regn. No. 2K21/PHDIT/504)

This is to certify that the above statement made by the candidate is true to the best of our knowledge and belief.

Dr. Virender Ranga
(Supervisor)
Associate Professor
Department of Information Technology
Delhi Technological University, Delhi, India

Dr. Ritu Agarwal
(Joint-Supervisor)
Associate Professor
Department of Information Technology,
Delhi Technological University, Delhi, India

Place: Delhi, INDIA

Date: 27/05/2025

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

Certificate

It is to certify that the work contained in this thesis entitled "**Design and Development of Security Solutions in Blockchain**" submitted by Anita Thakur (2K21/PHDIT/504) for the award of the degree of **Doctor of Philosophy** at Delhi Technological University, Delhi, India contains original research work conducted under our guidance and supervision.

We attest that Ms. Anita Thakur has successfully met all the stipulated requirements and standards for the submission of the thesis. The research work presented in this thesis is original, as per our best knowledge, and has not been utilized as a foundation for the conferral of any other degree or similar recognition.

We, as supervisor and joint supervisor, affirm the authenticity and originality of the research work conducted by Ms. Anita Thakur and presented in this thesis.

Dr. Virender Ranga
(Supervisor)
Associate Professor
Department of Information Technology
Delhi Technological University, Delhi, India

Dr. Ritu Agarwal
(Joint-Supervisor)
Associate Professor
Department of Information Technology,
Delhi Technological University, Delhi, India

Place: Delhi, INDIA
Date: 27/05/2025

Dedicated to our deities and my family

ACKNOWLEDGMENT

The completion of this doctoral thesis has been a challenging yet incredibly rewarding journey. With profound gratitude and appreciation, I take this opportunity to acknowledge all those who have contributed, in various capacities, to the success of this work.

I would first like to express my deepest gratitude to my thesis supervisor, **Dr. Virender Ranga**, and Joint-Supervisor **Dr. Ritu Agarwal** whose continuous guidance, insightful feedback, and unyielding support have been invaluable to the development of this thesis. Their expertise and dedication to my academic progress have inspired me. I am grateful for the countless hours they have spent reviewing my work, providing constructive criticism, and pushing me to reach higher standards. Their encouragement and belief in my abilities, even during the most challenging times, have been crucial in completing this thesis.

I would like to thank our Head of Department and DRC Chairperson, **Prof. Dinesh Kumar Vishwakarma** for his support. I also want to extend my heartfelt thanks to our former Head of Department **Prof. Kapil Sharma** for his time, thoughtful critiques, and unwavering support. His constructive feedback has significantly improved the quality of this research, and I have learned much from his expertise.

On a personal note, I want to express my heartfelt appreciation to my family for their unwavering love, patience, and encouragement throughout this journey. To my parents **Mr. Sunder Singh Thakur** and **Mrs. Suhari Devi**, thank you for your constant emotional support and understanding and for always being there when I needed you most. Your sacrifices and belief in my potential have been a continuous source of motivation. To my siblings **Preety Thakur**, **Rajeev Thakur**, and **Monika Thakur**, thank you for your encouragement and for providing a sense of balance and normalcy during times of stress.

I want to express my deepest gratitude to my niece **Aayat Thakur** and nephews **Ribansh Thakur** and **Vinayak Thakur** who, with their innocent conversations during challenging times, provided me with much-needed comfort.

My gratitude goes to my friend **Abhishek Singh** for his intellectual and emotional support throughout the research process. His collaborative spirit, insightful discussions, and exchange of ideas have enriched this research journey.

Lastly, I would like to thank all those whose names may not have been mentioned but who have contributed to the success of this thesis in one way or another. Whether through intellectual engagement, moral support, or acts of kindness, your contributions have been deeply appreciated.

MM,DD, YYYY Anita Thakur

ABSTRACT

In an era of increasing concerns over data privacy, security breaches, and fraud, blockchain offers a solution that empowers individuals and organizations to control their data and transactions. The transparency and immutability of blockchain records foster accountability and can significantly reduce the risk of fraud. Moreover, blockchain has the potential to enhance operational efficiency across industries. The medical field has increasingly embraced blockchain technology to address challenges related to critical data security, patient privacy, and operational efficiency. As healthcare systems become digitized, the need for robust mechanisms to protect sensitive patient information has never been more pressing. Despite the potential for improved security offered by blockchain, it has its own vulnerabilities. A significant issue is the occurrence of vulnerability in smart contracts, which can result in unforeseen exploits and considerable financial losses, as evidenced by events such as the DAO hack. This thesis delves into integrating blockchain in healthcare, addressing critical issues such as data privacy, security, access control, and scalability. It proposes solutions to enhance data security and privacy in healthcare systems by examining various existing studies and frameworks.

The thesis makes a substantial **contribution** in designing security solutions, performing extensive performance evaluations of blockchain networks, and developing a healthcare framework that leverages blockchain technology as follows:

[1] An extensive systematic literature review has been undertaken to investigate the various challenges associated with adopting blockchain solutions within the healthcare industry. This analysis underscores the intricate challenges associated with incorporating blockchain into current healthcare frameworks, encompassing concerns regarding interoperability, data security, privacy, and adherence to regulatory standards. Additionally, it highlights the necessity for increased empirical investigations that evaluate blockchain technologies' practical implementation and efficacy across diverse healthcare scenarios, including patient data management, supply chain integrity, and clinical trials. Alongside identifying these gaps, the review also highlights the fundamental challenges that blockchain technology may pose.

[2] As organizations and developers adopt blockchain solutions for various applications, it is essential to assess the performance of these platforms to inform decisions that align with their requirements and use cases. This research work analyzes the performance of several popular blockchain platforms to fill a gap in the literature.

[3] It contributes to the smart contract vulnerability detection field by employing advanced machine learning techniques to develop a multi-label classification model with an impressive accuracy of 95.05% and a detection time of 0.11 seconds. By leveraging a diverse dataset of known vulnerabilities, the adopted approach not only identifies multiple vulnerabilities within a smart contract but also enhances the efficiency of detection process, addressing the limitations of traditional expert-defined criteria.

[4] To ensure security and access control, an efficient cryptographic scheme is proposed. Conventional access-control mechanisms in blockchain encounter data ownership and management difficulties, particularly when data owners lack trust in storage providers. However, encryption before uploading offers some advantages. Traditional techniques such as Elliptic Curve Cryptography (ECC), RSA, and Advanced Encryption Standard (AES) face key management and precise control challenges. The proposed ciphertext policy attribute-based encryption mechanism allows data owners to establish access policies and encrypt information before uploading, thereby improving security in decentralized settings. The experimental results demonstrate the relative performance of the proposed scheme, showing a significant reduction in computational costs.

[5] Healthcare systems struggle with securing Electronic Health Records (EHRs) due to vulnerabilities in centralized databases, leading to potential cyber threats. This research proposes using Hyperledger Fabric (HLF) with attribute-based access control and AES-256 encryption to protect sensitive data. A distributed off-chain storage solution is integrated to address scalability issues for efficient data management.

The Performance evaluation, analysis, and experimental results indicate that the proposed solutions offer a viable and effective environment. Moreover, the comparative study demonstrates that the suggested approaches outperform the existing solutions.

Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms/Abbreviations

ABAC	Attribute-based Access Control
AA	Attribute Authority (Hospital)
AST	Abstract Syntax Tree
ABI	Application Binary Interface
BDHE	Bilinear Diffie-Hellman Exponent
CP-ABE	Ciphertext-policy attribute based encryption
CA	Certificate Authority or Central Authority
CFG	Control Flow Graph
CSG	Code Semantic Graph
CNN	Convolutional Neural Network
CID	Content Identifiers
GNN	Graph Neural Network
GDPR	General Data Protection Regulation
DO	Data Owner (Patient or Health data owner)
DU	Data User (Physician, Researcher, etc.)
DL	Deep Learning
DT	Decision Tree
DLT	Distributed Ledger Technology
EHR	Electronic Health Record
EVM	Ethereum Virtual Machine
FN	False Negative
FP	False Positive
HLF	Hyperledger Fabric

IPFS	InterPlanetary File System
KNN	K-Nearest Neighbor
LSSS	Linear Secret-Sharing Schemes
LightGBM	Light Gradient Boosting Machine
LSTM	Long Short-Term Memory
MSP	Membership Service Provider
ML	Machine Learning
mHealth	Mobile Health
NIST	National Institute of Standards and Technology
PHR	Patient Health Record
PHI	Personal Health Information
PBFT	Practical Byzantine Fault Tolerance
PoW	Proof-of-work
PoS	Proof-of-stake
PoA	Proof-of-authority
RF	Random Forest
RNN	Recurrent Neural Network
SUT	System under test
SPDL	Security Protocol Description Language
SMOTE	Synthetic Minority Oversampling Technique
SWC	Smart Contract Weakness Classification
SVM	Support Vector Machine
TPS	Transaction per second
TF-IDF	Term Frequency- Inverse Document Frequency
TN	True Negative
TP	True Positive
XGBoost	Extreme Gradient Boosting

ZKP	Zero-Knowledge Proof

List of Symbols

$T_{Latency}$	Transaction Latency
$\widehat{d}_{t,u}$	Predicted value for u^{th} label
t	Given label
$d_{t,u}$	Corresponding true value
S_n	Number of samples
L	Number of labels
tl	Ground truth label set
pl	Predicted label set
$L_{hamming}$	Hamming Loss
H_{prev}	Hash of previous block
T	Timestamp
N	Nonce
D	Transaction Data
H_i	Hash of parent node
H_{right}	Hash of right child
H_{left}	Hash of left child
$\mathfrak{p}$	Probability of legitimate node mining the block
$\mathfrak{q}$	Probability of the attacker mining new block before legitimate node
Q_z	Probability that attackers catch up with honest nodes from z blocks
V_t	Target value
$\mathcal{D}$	Target difficulty

$\mathfrak{A}$	Adversary
$\mathfrak{C}$	Challenger
λ	Security parameter
$\mathbb{G}_1, \mathbb{G}_T$	Multiplicative cyclic group of prime order p
p	Prime order
g	Generator of G_1
e	Mapping function $e(G_1, G_1) \rightarrow G_T$
$\mathcal{H}$	Collusion resistant hash function
PP	Public Parameter
GPP	Global Public Parameter
MK	Master Key generated by the CA
uid_k	Unique identifier for user k
$Certificate(uid_k)$	User's certificate
$UPK_{uid_k}, UPK'_{uid_k}$	User Public Key
$USK_{uid_k}, USK'_{uid_k}$	User Secret Key
MSK	Master Secret Key generated by AA
PK	Public Key generated by AA
$\mathcal{U}$	Universe of attributes controlled by AA
VK_x	Version Key of the attribute
AK_x	Public Attribute Key of the attribute
S	Attribute sets of user
SK	Decryption Key for DU generated by AA
$\mathbb{A}$	Access Structure
CT	Ciphertext
$\widehat{x}$	Revoked attribute from user
$AUK_{\widehat{x}}$	Attribute Update Key
$KUK_{\widehat{x},k}$	user Key Update Key

$CUK_{\widehat{x}}$	Ciphertext Key Update
Ehr	Patient Record
DR_n	Doctor Record
PH_n	Pharmacy Record
LB_n	Lab Record
INS_n	Insurance Record
P_{ID}	Unique Identity of the patient/ PatientID
D_{ID}	Doctor ID
Ph_{ID}	Pharmacy ID of the associated patient
L_{key}	LabKey of associated patient/ Lab ID
Ins_{Key}	InsuranceKey of associate patient/ Insurance ID
P_{name}	Patient Name
D_{name}	Doctor Name
DOB	Date of Birth
$Gender$	Gender
$Addr$	Address
$PhNo$	Phone Number
Ins	Insurance
Med	Medication
$Diag$	Diagnosis
$Test_{Tn}$	Tests Taken
D_{Spec}	Doctor Specialization
D_{Work}	Doctor Days of Working
C_{at}	Timestamp at pharmacy report of patient created
U_{at}	Updated At
L_{test}	Lab Tests
T_{status}	Status of Lab test

Doc_{ID}	Insurance DocumentID of Patient
Adh_{No}	Adhaar Number of Patient
Amount	Insurance Amount patient claimed
Status	Status if the amount claimed accepted or rejected

Chapter 1

Introduction

The purpose of this chapter [*] *is to provide background information about blockchain technology and provides an overall introduction of the thesis. Also, the chapter explains the motivation, aim, objectives, and structure of the thesis.*

1.1 Background

The advent of Bitcoin in 2008 introduced the world to a groundbreaking technology known as blockchain. Blockchain is characterized by its cryptographic security, append-only structure, immutability, and the requirement for consensus among participants for updates [1]. In essence, blockchain is a decentralized, secure, and transparent public ledger that operates without a single point of failure.

Before delving into further specifics, the history of the blockchain is reviewed, as illustrated in Figure 1.1. In 1990, researchers Stuart Haber and W. Scott Stornetta [2] presented a data structure designed to link a series of temporally ordered messages at the CRYPTO conference. By 1991, they had articulated the concept of a cryptographically secured chain of blocks, laying the foundational ideas for blockchain, which began to take shape in the late 1980s and early 1990s.

In 2008, Satoshi Nakamoto [**?**] published a white paper that established the framework for blockchain. This document proposed a decentralized, peer-to-peer digital payment system utilizing public key cryptography, enabling direct online transactions between users without financial intermediaries. A significant contribution of Nakamoto's work was the solution to the double-spending problem, which had plagued earlier digital currency attempts.

[*]The part of this chapter has been published/accepted/communicated in: Anita Thakur, Virender Ranga and Ritu Agarwal, "Exploring the Transformative Impact of Blockchain Technology on Healthcare: Security, Challenges, Benefits, and Future Outlook," *Transactions on Emerging Telecommunications Technologies*, Wiley, vol. 36, p. e70087, 2025, DOI: `https://doi.org/10.1002/ett.70087`. **(SCIE, IF=2.5)**.

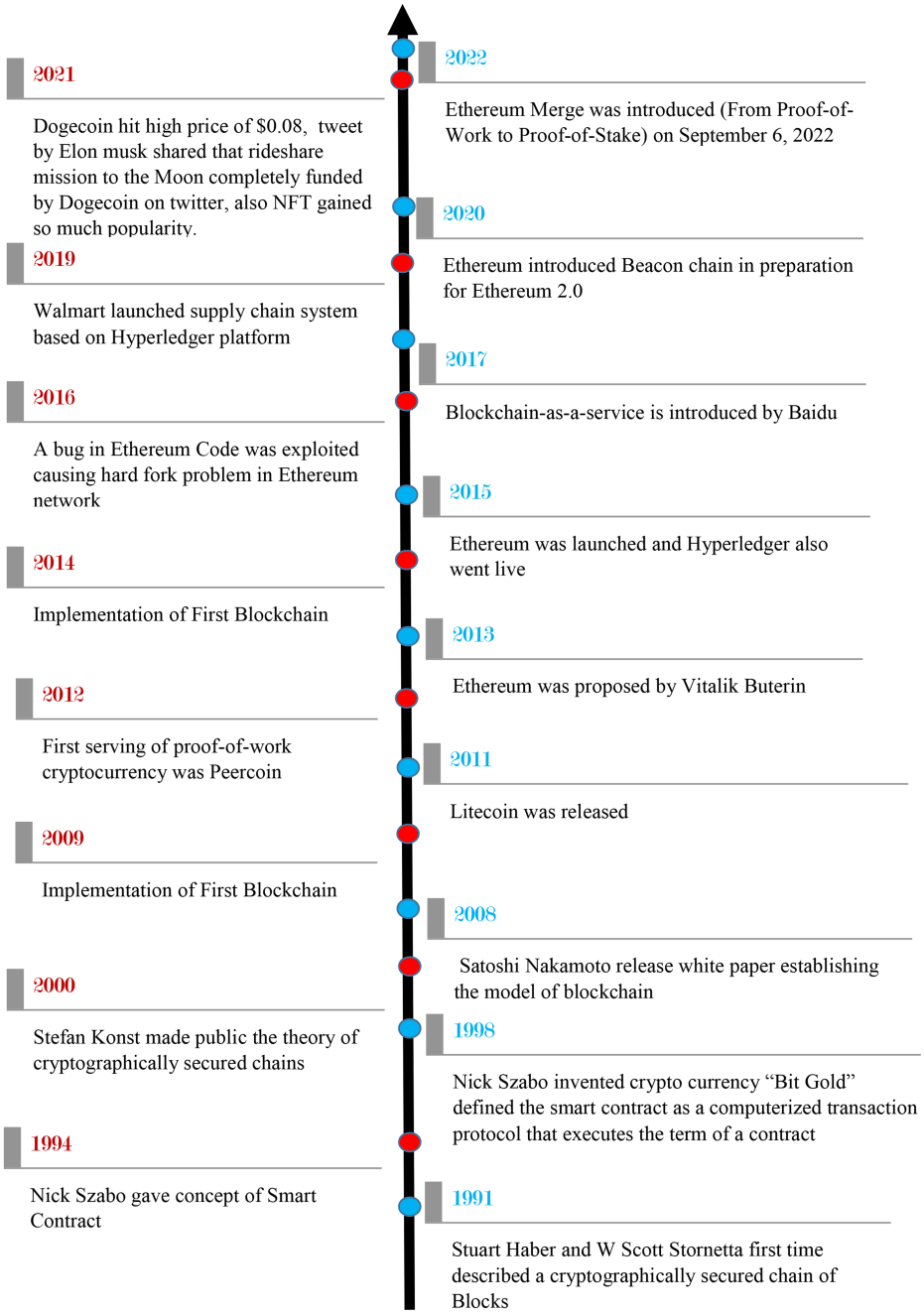


Figure 1.1: Key milestones in blockchain history

Nakamoto introduced a peer-to-peer network that employed a proof-of-work mechanism to create a public transaction history, making it computationally impractical for malicious actors to alter the data.

Each node maintains a copy of the blockchain, ensuring transparency and redundancy. Transactions are grouped into blocks, which are cryptographically linked to form a chain, making it nearly impossible to alter past records without consensus from the majority of nodes. It is evident that the fundamental responsibilities of a blockchain node include 1

Establishing a connection to the blockchain network, 2 Maintaining an up-to-date ledger, 3 Listening to transactions, 4 Transmitting valid transactions into the network, 5 Listening for newly created blocks, 6 Validating newly created blocks—confirming transactions , 7 Creating and transmitting new blocks

1.2 Motivation of Thesis

The blockchain field is rapidly evolving, and while many aspects of the technology have been developed and implemented, several open challenges still need attention. Blockchain offers solutions to numerous security, privacy, data immutability, and integrity issues. However, it also presents challenges, including scalability, computational complexity, energy consumption, and ensuring fine-grained data access for users within the network. Scalability is one of the primary concerns, as numerous networks encounter difficulties processing an increasing volume of transactions in a timely and efficient manner, resulting in longer transaction times and increased fees. Also, certain consensus mechanisms, such as Proof of Work, utilize significant energy, raising environmental concerns. Security vulnerabilities also introduce risks, as they can result in fraud or breaches due to vulnerabilities in network protocols or smart contracts.

1.3 Problem Statement

Blockchain offers improved security and transparency over centralized systems; nevertheless, its real-world application faces considerable performance hurdles, such as slow transaction processing speeds and scalability difficulties. Various existing research has largely concentrated on theoretical frameworks, highlighting the necessity for practical solutions that tackle transaction volume and processing speeds. It is important to identify the bottleneck and factors affecting blockchain performance in order to use blockchain in developing enterprise solutions. The performance of blockchain also depends on consensus algorithms, which are crucial for network efficiency and security.

Smart contracts are crucial in regulating blockchain operations by automating processes, enforcing agreements, ensuring transparency, and offering a secure and efficient framework for transactions and interactions between parties. However, the smart contract has several common vulnerabilities, including unchecked calls, reentrancy attacks, arithmetic errors, and front-running, which expose it to potential attacks. Smart contracts, unlike conven-

tional programs, cannot be modified after deployment and cannot be fixed for flaws. Most existing vulnerability detection solutions rely on expert-defined criteria, which are inefficient and have limited scalability. Blockchain is becoming increasingly common across various sectors, with healthcare being one of the most prominent areas. However, blockchain-enabled healthcare still faces critical challenges in maintaining the privacy and security of medical data, as the increasing volume of sensitive patient information shared across platforms raises the risk of data breaches, unauthorized access, and compromised data integrity. Specifically, the challenge lies in implementing effective fine-grained access control mechanisms that protect sensitive patient information while ensuring authorized users can access the data as needed.

1.4 Research Objectives

The primary objectives of this thesis are as follows:

[1] To evaluate the performance and scalability of different existing blockchain platforms.

[2] To design and develop security solutions for data transmission in blockchain.

[3] To propose a new framework for access control and enhanced scalability of blockchain leveraging solutions.

1.5 Contribution of Thesis

This thesis focuses on developing advanced solutions to guarantee the security, privacy, and integrity of data. Sensitive information such as health records that include EHR, PHR, PHI, and mHealth data need to be stored and accessed in a way that only the authorized individual can perform the authorized operation. A new and efficient approach, BloCPABE is proposed which integrates CP-ABE into blockchain. This solution is designed to provide comprehensive protection by supporting efficient data access control. Also, to address the challenges of efficient data sharing in healthcare, this thesis introduces the ABHealChain system that employs permissioned blockchain along with ABAC, enabling secure and controlled data sharing among authorized entities.

The significant contribution of this thesis is demonstrated below:

[1] The literature review evaluates current healthcare solutions incorporating blockchain, emphasizing their security, privacy, benefits, and challenges. It also highlights the

benefits and drawbacks of blockchain, offering a detailed view of the practical challenges that need to be addressed to realize its complete potential. It enhances the understanding of various applications of blockchain and the security challenges linked to its integration in healthcare.

[2] To make a well-informed decision on selecting a blockchain platform for implementing a blockchain-enabled solution, the performance of HLF, Ethereum (private deployment), and Hyperledger Besu is evaluated. The findings highlight HLF's superior performance across all measures. Also, an extensive evaluation of HLF is conducted to analyze how factors like workload, TPS, and organizational elements impact scalability and performance. Additionally, the research investigates common errors leading to transaction failures.

[3] An efficient approach named SmarConTest is introduced for the multi-label classification of smart contracts using opcodes. Present approaches are limited to binary detection of contracts, allowing for identifying only a single type of vulnerability. In contrast, SmarConTest can simultaneously identify multiple vulnerabilities within smart contracts. Our approach performed well on the test set, with $\approx 95.05\%$ accuracy for each vulnerability and achieving prediction time ≈ 0.11 second.

[4] A secure and expressive data access control method named BloCPABE is introduced that addresses mHealth security issues. BloCPABE uses Blockchain-based CP-ABE for efficient attribute revocation, protecting user privacy and security from attacks. The technique provides backward and forward security and immutable, tamper-proof data integrity via blockchain.

[5] A blockchain-enabled healthcare solution that utilizes HLF in conjunction with attribute-based access control and AES-256 encryption to safeguard sensitive data is introduced. A distributed off-chain storage system is implemented to resolve scaling challenges and enhance data management efficiency.

[6] Simulations are conducted to test the Raft and PBFT consensus algorithms in blockchain networks. The performance is measured by evaluating throughput, packet delivery ratio, packet loss ratio, leader election time, and agreement time using the NS3 network simulator.

[7] The formal verification of the proposed attribute-based encryption scheme is conducted using verification tools such as AVISPA SPAN and the Scyther tool. These

tools simulate the presence of an active intruder, enabling the interactive identification and construction of potential attacks against various protocols.

[8] The security and performance of the proposed solutions are evaluated through formal and experimental analysis, showcasing their capabilities in a healthcare setting.

1.6 Thesis Organization

This section outlines the structure of the thesis, which consists of eight chapters as detailed below:

Chapter 1: Introduction

This chapter provides the basic information about the blockchain. This chapter outlines the motivation for the thesis, the problem statement, the research objectives, and the contributions of the thesis. This chapter also includes a concise outline of the thesis.

Chapter 2: Literature Review

The literature review section provides an extensive analysis of studies addressing the implications of workload dynamics on the performance and scalability of different existing blockchain networks, the security and privacy preservation of data within blockchain systems, and the transformative role of blockchain in the healthcare sector. This chapter outlines the research methodology by clearly defining and elucidating the research problem, supported by formulated research questions. This chapter concludes by identifying research gaps in the existing literature.

The paper listed below is published as part of this research:

[1] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Exploring the Transformative Impact of Blockchain Technology on Healthcare: Security, Challenges, Benefits, and Future Outlook," *Transactions on Emerging Telecommunications Technologies* , Wiley, vol. 36, p. e70087, 2025, DOI: `https://doi.org/10.1002/ett.70087`. **(SCIE, IF=2.5)**.

Chapter 3: Preliminaries

This chapter presents foundational insights into the fundamental principles of blockchain technology. The chapter explains the technical foundations of blockchain, examines its various applications across multiple sectors, and emphasizes its role within the healthcare

industry. It also addresses the fundamental elements utilized in developing attribute-based encryption methods.

Chapter 4: Implications of Workload Dynamics on Blockchain Scalability and Performance

This chapter analyzes the effects on performance and scalability when subjected to diverse workloads across various blockchains. Additionally, offering comprehensive observation assists in making a well-informed decision regarding a suitable platform for implementation. The discussion includes the different errors that contribute to transaction failure.

The papers listed below are published as part of this research:

[1] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Workload dynamics implications in permissioned blockchain scalability and performance," *Cluster Computing*, Springer, vol. 27, issue. 8, pp.11569-11593, 2024, DOI: `https://doi.org/10.1007/s10586-024-04550-z`. **(SCIE, IF=3.6)**.

[2] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Performance benchmarking and analysis of blockchain platforms", *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, Delhi, pp. 1-7.

Chapter 5: Smart Contract Vulnerabilities Detection using Machine Learning

This chapter introduces an efficient smart contract vulnerability detection approach known as SmarConTest. SmarConTest employs multi-label classification for smart contract vulnerability detection, as a single contract has more than one vulnerability. Common vulnerabilities include reentrancy attacks, integer overflow/underflow, access control issues, etc.

[1] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "SmarConTest: An Efficient Smart Contract Vulnerability Detection Method Using Machine Learning,"—communicated

[2] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "A Review On Smart Contract Vulnerability Detection Using Deep Learning", —communicated

Chapter 6: Revocable and Privacy-Preserving Data Access Scheme in Blockchain

This chapter introduces a ciphertext policy attribute-based encryption scheme that ensures the proposal of a revocable and secure fine-grained access mechanism utilizing blockchain

and CP-ABE. It outlines a robust and decentralized framework for data sharing that combines attribute-based encryption with IPFS and blockchain, enabling precise access control over encrypted data.

[1] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Revocable and Privacy-Preserving CP-ABE Scheme for Secure mHealth Data Access in Blockchain," *Concurrency and Computation: Practice and Experience*, Wiley, vol. 37, p. e70064, 2025, DOI: `https://doi.org/10.1002/cpe.70064`. **(SCIE, IF=1.5)**.

[2] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Attribute-based encryption scheme for secure and efficient access in blockchain", *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, Bangalore, pp. 653-658.

Chapter 7: Blockchain-Enabled Healthcare System with Attribute-based Access Control

This Chapter presents an architectural framework designed to enhance the privacy and security of patient data sharing across different healthcare entities while tackling the challenges mentioned above, such as security, privacy, scalability, etc. The framework utilizes HLF for its implementation. Additionally, it integrates mechanisms for attribute-based access control (ABAC) to guarantee restricted access to sensitive patient information, with a primary emphasis on maintaining the system's integrity, confidentiality, and authenticity, especially for designated healthcare entities.

[1] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "ABHealChain: Enhancing Privacy and Security in Healthcare Data Sharing through Hyperledger Fabric and Attribute-based Access Control",—communicated.

[2] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Simulation-based Performance Evaluation of Consensus Algorithms in NS3 for Blockchain Network", *International Conference on Futuristic Technologies (INCOFT2025)*, Pune.

Chapter 8: Conclusion and Future Scope

This final chapter outlines the findings and potential directions for future research inquiries. This chapter examines the significance of proposed solutions in enhancing security and safeguarding privacy.

List of Publications : This section compiles papers published, accepted, or communicated in SCIE Journals and Scopus-Indexed Conferences.

References: This section compiles sources cited throughout this thesis.

In the following chapter (Chapter 2), we present a comprehensive review of the existing literature related to security, privacy, and access control challenges in blockchain systems, along with mitigation strategies proposed in recent research. Additionally, the chapter explores application of blockchain technology in the healthcare domain, highlighting its potential roles, associated benefits, and prevailing challenges in healthcare.

Chapter 2

Literature review

The objective of this chapter * *is to provide an exhaustive literature review on security, privacy, and scalability challenges in the blockchain and blockchain-enabled solutions. This chapter delineates the methodology employed in performing the literature review, articulates the research questions, pinpoints existing research gaps, and outlines prospective ways for future research based on the findings.*

2.1 Introduction

The blockchain, a distributed ledger network, is based on the principle of permanent record addition and prohibits any alterations without unanimous consent [4]. Blockchain is an effective platform that has the potential to apply in a variety of industries. Among the numerous blockchain applications, healthcare has the potential to show off the capabilities of blockchain. The healthcare system's quality is significantly enhanced by sharing of data

*The part research work covered in this chapter has been published/accepted/communicated in: Anita Thakur, Virender Ranga and Ritu Agarwal, "Exploring the Transformative Impact of Blockchain Technology on Healthcare: Security, Challenges, Benefits, and Future Outlook," *Transactions on Emerging Telecommunications Technologies* , Wiley, vol. 36, p. e70087, 2025, DOI: https://doi.org/10.1002/ett.70087. **(SCIE, IF=2.5)**
&
Anita Thakur, Virender Ranga and Ritu Agarwal, "Workload dynamics implications in permissioned blockchain scalability and performance," *Cluster Computing*, Springer, vol. 27, issue. 8, pp.11569-11593, 2024, DOI: https://doi.org/10.1007/s10586-024-04550-z. **(SCIE, IF=3.6)**
&
Anita Thakur, Virender Ranga and Ritu Agarwal, "Revocable and Privacy-Preserving CP-ABE Scheme for Secure mHealth Data Access in Blockchain," *Concurrency and Computation: Practice and Experience*, Wiley, vol. 37, p. e70064, 2025, DOI: https://doi.org/10.1002/cpe.70064. **(SCIE, IF=1.5)**
&
Anita Thakur, Virender Ranga and Ritu Agarwal, "ABHealChain: Enhancing Privacy and Security in Healthcare Data Sharing through Hyperledger Fabric and Attribute-based Access Control", —communicated.
&
Anita Thakur, Virender Ranga and Ritu Agarwal, "SmarConTest: An Efficient Smart Contract Vulnerability Detection Method Using Machine Learning," —communicated

and the availability of medical data to the authenticated healthcare providers, researchers, patients, and other consumers [5]. Security vulnerabilities within the blockchain ecosystem can result in substantial repercussions. They can modify data, facilitate double spending, result in financial theft, impede network operations, and, most importantly, undermine trust in the technology [6].

2.2 Approach

Our review process comprised three primary stages: planning, reviewing, and reporting. In planning phase, clear research objectives and questions are established. Subsequently, the reviewing stage involved the selection of pertinent articles based on a well-defined selection. Finally, the findings of this thesis are comprehensively reported, as illustrated in Figure 2.1.

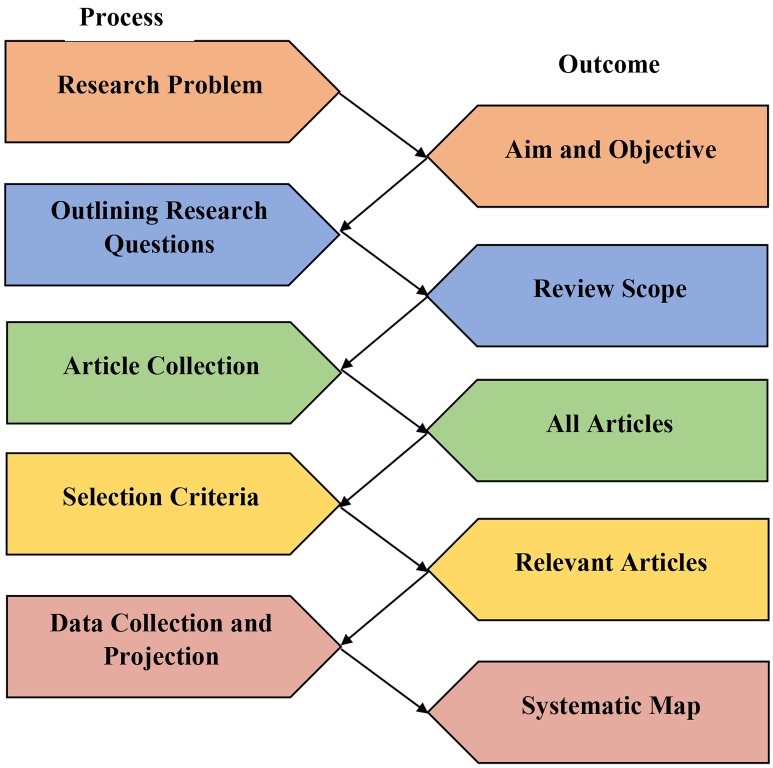


Figure 2.1: Literature review development guide

2.2.1 Planning Phase

The planning phase is pivotal in formulating clear research questions that address critical issues related to security, privacy, data access, and scalability within blockchain and

blockchain-enabled solutions. The significance of this phase cannot be overstated, as it sets the direction for subsequent stages of the research, including the literature review and data analysis.

Research questions (RQ)

[1] **RQ1 :** What is the impact of increased workload on the scalability and performance of the blockchain?

[2] **RQ2 :** What are the security, privacy, scalability, and access control issues and their countermeasures in blockchain?

[3] **RQ3 :** What are the different strategies used for security and privacy-preservation?

[4] **RQ4 :** What is the role of blockchain in transforming the healthcare domain?

[5] **RQ5 :** How does blockchain handle the security issues of the healthcare system?

[6] **RQ6 :** What are the benefits and challenges of incorporating blockchain in healthcare?

2.2.2 Reviewing Phase

In this phase, an analytical approach is applied to identify the areas where more research is required. Once relevant articles are identified, they undergo a meticulous evaluation process. The articles are critically assessed using clearly defined inclusion and exclusion criteria to ensure their relevance to the research topic, as described in Table 2.1.

This rigorous filtering process helps select studies that meet methodological rigor, data integrity, and scholarly contribution standards. The prior studies specified in languages other than English are excluded. Various criteria are applied for the refinement of the survey, such as excluding duplicate papers based on titles, abstracts, and documents that do not satisfy the requirements of the study.

2.2.3 Reporting Phase

This phase summarizes the findings related to the research questions. The reporting phase aims to communicate the conclusions drawn from the analysis, reflecting the relevance and implications of the findings in the field of blockchain, especially in the context of security, privacy, scalability, and healthcare applications.

Table 2.1: Inclusion and exclusion criteria

Criteria	Eligibility Criteria
Inclusion	Studies that address security/privacy issues in blockchain and blockchain leveraging solutions
	Studies addressing security solutions in blockchain
	Benefits and challenges of blockchain
	Performance analysis of blockchain platforms and blockchain-based healthcare systems
	Blockchain in healthcare and mHealth
	Smart contract vulnerability
	Access-control mechanism in blockchain
	Solutions to scalability of blockchain
Exclusion	Studies that are conducted in a language other than English
	Studies not relevant to the selected domain

2.2.3.1 What is the impact of increased workload on the scalability and performance of the blockchain?

The primary objective of this **RQ1** is to establish the architectural and conceptual underpinnings for implementing blockchain in various sectors. Nevertheless, the successful implementation of blockchain technology requires a meticulous assessment of variables such as transaction volume, transaction rates, and the number of participants. These variables have the potential to considerably impact performance, including throughput, latency, resource consumption, and success rate.

The study [7] demonstrates that HLF v1.0 outperforms v0.6 regarding scalability, execution time, latency, and throughput when the workload and number of nodes are adjusted. The research indicates that HLF v1.0 outperforms v0.6 in sustaining consistent performance across various metrics, irrespective of the network size. This study does not examine the impact of fundamental consensus mechanism on these findings.

Another study by Shalaby et al. [8] assesses the performance of HLF v1.4 by analyzing the influence of a variety of configurable parameters, including the number of endorsing peers, batch size, and batch expiration, on end-to-end transaction latency and network throughput. The number of concurrent transactions increases, and the batch expiration is altered, which affects both throughput and latency. The efficacy and latency are balanced by setting the batch size to 45 transactions. Allowing all transactions to fit within a single block, this size prevents the maximum batch size from being reached before the expiration. Nevertheless,

the number of parallel transactions within the block tends to increase, resulting in latency, conversely, throughput benefits.

The HLF-GLDB benchmark utility is designed to simulate database access in HLF. Data compression in GoLevelDB is a performance impediment in the HLF network, as demonstrated by the results of [9]. The network's overall efficacy is enhanced by 54% when data compression is disabled. The study [10] demonstrates that the performance of HLF version 2.0 is considerably improved compared to previous versions of the blockchain framework. In [11], the approach of obtaining measurement data from the system under study to ascertain the pattern of resource utilization or performance degradation that could be caused by software aging is discussed.

Ethereum and HLF are blockchains that have been broadly recognized and have experienced substantial industry adoption. To compare the performance of these two blockchains, a study [12] evaluates them in a variety of workload environments and determines that HLF consistently outperforms Ethereum in the areas of throughput, latency, and execution time. Despite the advancements made by both platforms, they continue to fall short of the capabilities of contemporary database systems when managing heavy workloads. Nevertheless, Ethereum can process more concurrent transactions than HLF despite the identical computational resources.

The impact of scalability, bottlenecks, and ordering services in HLF has been challenging to understand due to the performance complexity of distributed systems. To resolve this matter, the authors [13] introduce an exhaustive assessment of HLF functionality and scrutinized the execution, ordering, and validation phases. Solo, Kafka, and Raft are the three procurement services that are compared in the review. The results indicated that the OR endorsement policy exhibited adequate scalability during the execution phase, while the AND endorsement policy did not perform satisfactorily.

2.2.3.2 What are the security, privacy, scalability, and access control issues and their countermeasures in blockchain?

This **RQ2** discusses security vulnerabilities such as identity theft and data breaches (Figure 2.2), privacy concerns related to data exposure, scalability issues that hinder transaction speed and network efficiency, and access control complexities that arise in managing permissions and users.

Mollajafari et al. [14] examine the security risks and vulnerabilities inherent in the seven-

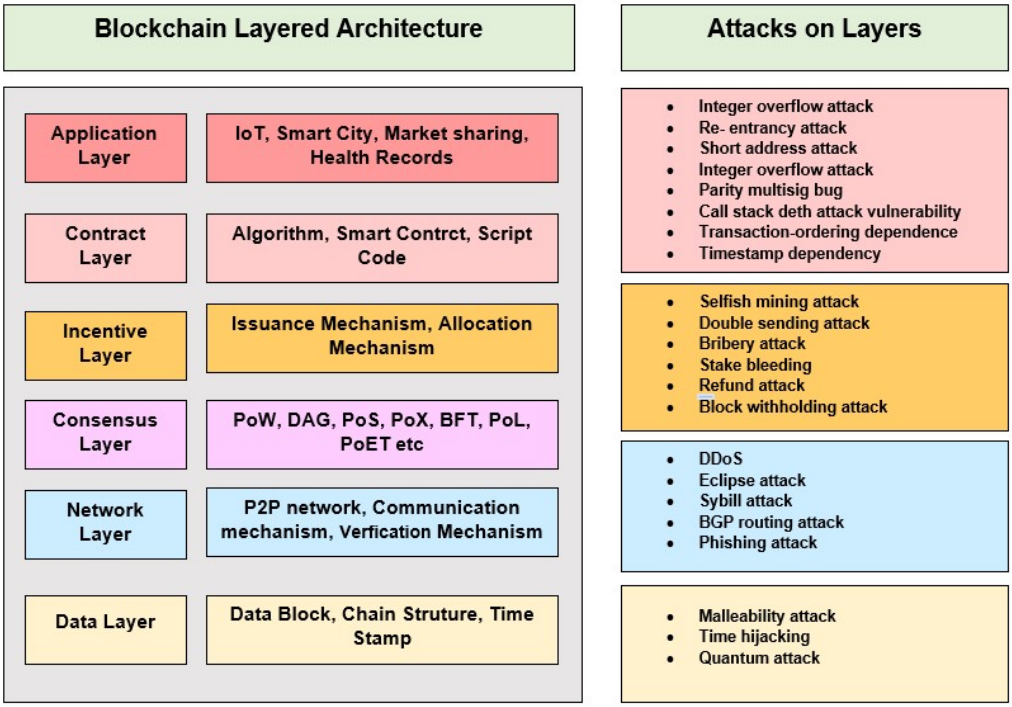


Figure 2.2: Attacks in layered architecture of blockchain

layer architecture of blockchain. Unique security challenges are presented by each layer, from the data layer, which concentrates on data integrity and confidentiality, to the presentation layer, which addresses user interaction and experience. Data manipulation in the data layer, denial of service, sybil attacks in the network layer, and 51% and long-range attacks in the consensus layer are among the most notable attacks. The incentive layer is vulnerable to economic exploits, while the contract layer risks integer overflow attacks and reentrancy. The application layer is susceptible to malware, while user interface deficiencies may affect the presentation layer. Cryptographic techniques, secure coding practices, robust governance mechanisms, and user education are all recommended as countermeasures to mitigate these risks.

Duan et al. [15] also investigate various security risks inherent to the Ethereum blockchain, with a particular emphasis on smart contracts. The study identifies a variety of attacks, such as denial-of-service attacks that can disrupt network operations and reentrancy attacks, which exploit the execution flow of smart contracts. Furthermore, the authors address vulnerabilities associated with inadequate access control and the potential for front-running attacks, which allow malicious actors to manipulate transaction ordering for personal advantage.

Given these security concerns, privacy challenges also emerge as a critical issue in blockchain

systems. Bernabe et al. [16] conduct a systematic review of the privacy challenges in blockchain, identifying problems resulting from the inherent transparency of public blockchains and the controlled environments of permissioned blockchain.The primary concerns are transaction linkability, conformance with privacy regulations such as the GDPR, and cryptographic key management. A variety of privacy-preserving strategies have been suggested to address these obstacles. Public blockchains utilize ring signatures, blending services, and ZKPs to obfuscate transaction details and improve anonymity. Conversely, permissioned blockchains employ anonymous credential systems and secure multi-party computation to ensure compliance and operational efficiency while maintaining privacy through controlled access and data segregation.

Xie et al. [17] identify significant challenges concerning throughput, storage, and networking that impede the performance of blockchain systems. Throughput constraints are apparent in widely used blockchains such as Bitcoin, which can execute merely seven transactions per second. Storage difficulties emerge from the necessity for each node to preserve a comprehensive transaction history, which can be onerous in resource-limited settings. Networking problems arise from the traditional broadcast method, resulting in bandwidth usage and delays in block propagation. Diverse solutions have been suggested to tackle these difficulties, such as augmenting block size, minimizing transaction size through methods like SegWit, and executing off-chain transactions via networks like the Lightning Network. Moreover, sharding, integrating blockchain with distributed storage systems, and enhancing data transmission protocols seek to optimize scalability while maintaining a balance among decentralization, security, and immutability.

Access control in blockchain pertains to regulating who can access and engage with data recorded on the blockchain. Challenges frequently stem from unauthorized access, data privacy issues, and the necessity for efficient systems to grant and revoke permissions while ensuring transparency and security. ABAC, ABE, fine-grained access control, granular attribute-based, role-based, and auditable access control methodologies establish a comprehensive framework for protecting sensitive data while facilitating efficient access for authorized users [18], [19], [20].

Qian et al. [21] examine the security and dependability of smart contracts, a growing problem in blockchain technology. They categorize standard vulnerabilities inside Solidity code, EVM execution, and block dependency levels. The authors categorize smart contract vulnerability detection research into five distinct areas: formal verification, symbolic execution,

Table 2.2: Vulnerabilities in Ethereum blockchain and their description

Vulnerability	**Description**
Reentrancy Vulnerability	This attack happens when a contract invokes an external contract prior to modifying its internal state. This sequence enables the external contract to initiate further calls back to the original contract, potentially leading to unforeseen results.
Access Control Vulnerability	This vulnerability in smart contracts is a security flaw that allows unauthorized individuals to gain access or alter the contract's data or functions.
Gas Limits and Loops	smart contract transactions can fail if the gas consumed exceeds the block gas limit. It often occurs in loops processing dynamic data, where the number of iterations is unpredictable and can potentially consume excessive gas.
Timestamp Dependence	When an smart contract depends on the block timestamp the blockchain network provides, a timestamp dependence issue occurs.
Integer Underflow and Overflow	When mathematical operations exceed the storage limitations of a data type, they can result in integer overflow and underflow, which might have unexpected consequences.
Front running	Front running attack occurs when a malevolent actor watches a transaction before it is verified and then submits a transaction that exploits the data in the observed transaction. It results in a number of attack patterns that are commonly known as insertion, displacement, and suppression attacks.
ERC-20 Token Issues	ERC-20 tokens are digital assets formed on the Ethereum blockchain, enabling developers to create tokens without creating a new chain. However, its basic implementation contains a critical security flaw that could result in the irreversible loss of funds for users.
Self-Destruct Vulnerability	The self-destruct vulnerability in smart contracts refers to the potential risks associated with the SELFDESTRUCT opcode, which allows a contract to be permanently deleted and its Ether balance sent to a specified address. Misuse or exploitation can lead to loss of funds or unintended contract behavior.
Denial of Service	It refers to a vulnerability in which an smart contract becomes inoperable due to improper handling of transactions that fail or are deliberately reverted. It can prevent users from interacting with the contract as intended, effectively rendering it unusable.
Frozen Ether	Frozen Ether vulnerability occurs when an smart contract allows the receipt of Ether but lacks mechanisms (like call, send, or transfer) to withdraw it, effectively locking funds. This issue can arise from coding flaws or accidental actions, as seen in the Parity Wallet incident, where millions in Ether became inaccessible.
Dangerous Delegatecall	Delegatecall allows a contract to run code from a different contract while maintaining its own context. This functionality can introduce security risks, especially when arbitrary addresses are permitted, as it could allow attackers to alter state variables or run harmful code.
Tx.origin	It is a global variable in Solidity that refers to the original address that initiated the transaction, even if the transaction involved multiple contracts. If tx.origin is used to check ownership of a contract, an attacker could trick a contract into performing actions on their behalf by calling it through a chain of contracts.

fuzzing detection, intermediate representation, and deep learning. The study evaluates accuracy, F1-score, and average detection time over 300 authentic Ethereum smart contracts.

The authors address vulnerability identification challenges and propose incorporating deep learning techniques in the next research.

Table 2.3: Various vulnerabilities detection tools

Tool	Method	Vulnerabilities											
		RE	TSD	AC	Gas	BR	U/O	TxO	SD	FE	DC	TOD	DoS
Oyente	Symbolic execution	✓	✓	✗	✓	✓	✓	✗	✗	✗	✓	✓	✗
Slither	Intermediate representation	✓	✓	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗
Manticore	Symbolic representation	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗
SmartDagger [22]	Static analysis	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓
SmartCheck [23]	Intermediate representation	✓	✓	✗	✓	✗	✓	✗	✗	✓	✗	✗	✓
Contractsentry [24]	intermediate representation	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
SafeCheck [25]	Intermediate representation	✓	✓	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓
Mythril	Symbolic execution	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗
Eth2Vec [26]	ML based static analysis	✓	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗
eTainter [27]	Code analysis	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓
DefectChecker [28]	Symbolic execution	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
EtherSolve [29]	Symbolic execution	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
sGUARD [30]	Symbolic execution	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗

RE: Reentrancy, TSD: Timestamp Dependence, AC: Access Control, Gas: Gas related vulnerability, BR: Bad Randomness, U/O: Integer Underflow/Overflow, TxO; Tx.origin, SD: Self-Destruct Vulnerability, FE: Frozen Ether, DC: Delegatecall, TOD: Transaction Order Dependence, DoS: Denial of Service

There are common vulnerabilities in smart contracts (refer to Table 2.2), including reentrancy attacks and arithmetic errors, rendering them appealing targets for adversaries. Due to the unchangeable nature of smart contracts, rectifying vulnerabilities after deployment is especially difficult. The overview of the vulnerability detection solutions evaluated in the previously mentioned literature is provided in Table 2.3. The literature study underscores the necessity of prioritizing multi-label detection strategies, as various smart contracts may exhibit multiple vulnerabilities simultaneously, a situation that traditional single-label methods cannot fully handle.

2.2.3.3 What are the different strategies used for security and privacy preservation?

The examination of current literature on **RQ3** uncovered diverse viewpoints concerning the format of the initial input data employed in vulnerability detection. The input data generally comprises the source code, opcode, bytecode of the contracts, pictures, or a mix thereof. Diverse detection methodologies depend on distinct representations of contract code to ascertain potential flaws. Typical input formats for these tools encompass Control Flow Graphs (CFGs), unprocessed Solidity code, Abstract Syntax Trees (ASTs), and symbolic execution traces.

To improve the precision of vulnerability identification, it is essential to delineate the characteristics of vulnerable source code accurately. The authors [31] introduce a smart contract vulnerability detection solution named Lightning Cat, which utilizes deep learning models for enhanced performance. It presents an efficient data pre-processing technique utilizing CodeBERT to extract semantic characteristics of vulnerabilities. Experimental findings indicate that Lightning Cat surpasses other tools referenced in [31].

Graph Neural Networks and expert-defined patterns are employed to identify vulnerabilities in smart contracts as detailed in the paper [32]. The technique depends on the source code of smart contracts to identify expert patterns, particularly for vulnerabilities. It designates program items as nodes and produces edges to signify their links and dependencies. It constructs a contract graph that encapsulates control flow and data flow semantics from the source code. The proposed method has demonstrated efficacy through comprehensive trials on VNT Chain and Ethereum smart contracts, identifying defects with high precision.

Duy et al. [33] present the VulnSense framework, which employs multimodal learning to detect vulnerabilities in Ethereum smart contracts. Source code, opcode sequences, and control flow graphs produced from the bytecode of smart contracts are amalgamated. The research [34] introduces a technique for identifying vulnerabilities in smart contracts through deep learning and multimodal decision fusion. This methodology employs three categories of data: the source code, which reveals the original logic and structure of the smart contract; the opcode, which encompasses low-level instructions for the Ethereum Virtual Machine; and CFG, which illustrates the control flow within the contract, emphasizing interactions between code segments and accessible execution paths. The method enhances vulnerability identification by amalgamating several kinds of data through a multimodal decision fusion strategy for superior feature extraction.

The study [35] examines AST-structured and program slice data. The author presents automated techniques to segment source code and pinpoint vulnerable sections. Structured information is generated by transforming source code into an AST, which hierarchically represents syntax and structure.

The paper [36] proposes smart contract source code and bytecode through the utilization of graph structures. It demonstrates the interconnections of control and data within source code using a Code Semantic Graph (CSG). Nodes signify program components such as function calls and variables, whereas edges denote control and data flow exchanges. Every edge possesses a temporal sequence that corresponds to its coded position. The authors extract

CFG from the bytecode, with nodes representing instruction blocks and edges denoting control flow connections.

The author [37] introduces DL5SC, a new DL framework for detecting vulnerabilities in smart contracts at the opcode level. The exploration of image input components presents a new method for improving the detection of vulnerabilities in SCs. The author [38]introduced a new DL method, named CodeNet, aimed at identifying vulnerabilities in smart contracts through a code-targeted CNN architecture focused on code analysis.

These advancements in detecting vulnerabilities highlight the importance of secure systems, not just in smart contracts but across various cryptographic schemes. For example, In the work of Chase, M. [39], a multi-authority CP-ABE (MCP-ABE) scheme is presented that leverages a central authority (CA). This CA possesses the capability to decrypt all ciphertexts within the system. This inherent decryption ability used by the CA introduces a vulnerability in the security position of decryption key storage.

Later, Chase et al. [40] introduce an MA-ABE scheme that operates without a centralized trusted authority. However, under this scheme, data consumers are required to obtain at least one attribute from each Attribute Authority (AA), thus limiting its practicality. In contrast, Lewko et al. [41] developed a CP-ABE scheme featuring fully decentralized multi-authorities, eliminating the necessity for attribute authorities to collaborate. Consequently, data consumers can acquire any desired number of attributes from any AA within this framework.

Yang et al. [42] address the critical challenge of data access control in cloud storage systems, particularly in the context of untrusted servers and the limitations of existing CP-ABE schemes. They propose a novel revocable multi-authority CP-ABE scheme that enhances expressiveness, efficiency, and security, allowing multiple authorities to independently issue attributes while ensuring effective attribute revocation with backward and forward security. However, the approach may face challenges related to the complexity of managing multiple authorities and the potential for increased overhead in communication and coordination. Additionally, while the proposed method improves efficiency by reducing the need for extensive ciphertext updates, it may still encounter scalability issues as the number of users and authorities grows.

To address the risks associated with semi-honest cloud servers, Miao et al. [19] propose a method that involves splitting secret keys associated with certain attribute sets. This

approach allows healthcare providers and patients to securely exchange sensitive health information while only ensuring access is restricted to authorized individuals. The implementation challenges concerning scalability and the mechanism for attribute revocation related to the study require discussion.

Data consumers can leverage third-party servers for decryption computations to mitigate the computational burden associated with the decryption phase. Tu et al. [18] introduce a multi-authority CP-ABE scheme supporting decryption outsourcing. In this system, data consumers do not have to do any complicated Bilinear pairing operations during the decryption phase. The security of ciphertext entrusted to a third-party custodian presents potential vulnerabilities to tampering, as documented by Guo et al. [43]. Blockchain mitigates this risk by demonstrably guaranteeing both data integrity and non-repudiation. These inherent cryptographic properties underpin the increasing adoption of blockchain across diverse application domains.

Wei et al. [44] propose a revocable hierarchical scheme, RS-HABE, to enhance system security and meet application requirements in resource-constrained environments. This scheme enhances the original ABE by integrating user revocation, secret key delegation, and ciphertext updating mechanisms. ABE has seen extensive application in cloud services owing to the comprehensive expressiveness of ABE policies. Traditional ABE schemes exhibit significant computational overhead and a deficiency in verifiability, resulting in reduced efficiency and security for cloud-based applications. Hou et al. [45] proposed a blockchain-based solution that enhances decryption efficiency and enables users to verify the accuracy of decryption conducted by external parties. This method provides a more secure and efficient means of managing access to data stored in the cloud.

The growing dependence on cloud computing for managing PHRs has required strong encryption techniques to maintain data confidentiality. Zhang et al. [46] presented an ABE scheme incorporating keyword search, effectively addressing key management and access control challenges by facilitating fine-grained access control. The authors [47] present an innovative access control scheme, PAFR-ABE, aimed at improving usability in dynamic healthcare environments. This scheme features a dynamic revocation system that optimizes the management of secret keys, eliminating the need for updates for users who have not been revoked.

2.2.3.4 What is the role of blockchain in transforming the healthcare domain?

This **RQ4** examines role of blockchain across different healthcare sectors. Blockchain facilitates access to patient health records, verification of medical prescriptions, secure data sharing among providers, drug provenance verification, safeguarding clinical trials, managing medical device supply chains, overseeing revenue management, and transferring funds. This research question encompasses critical areas of blockchain applications in healthcare, including patient health records, pharmaceutical supply chain management, clinical trial data management, and remote patient monitoring.

The PcBEHR system architecture [48] offers mechanisms for patients to manage and control their medical data. In PcBEHR, patients own their medical data and can grant access permissions. The patient-controlled blockchain-enabled EHR solution employs access control, privacy, and interoperability functionalities. MedRec [49] is a decentralized electronic medical record management system that employs blockchain technology to document sensitive information and operations, ensuring confidentiality, authentication, accountability, and data sharing for critical data. MedRec employs a smart contract to ensure confidentiality by regulating viewership rights. Patients are provided comprehensive access to their medical data from various providers consolidated in one location.

Chen et al. [50] developed a storage scheme that integrates blockchain and cloud storage to effectively manage patients' sensitive healthcare records. This system facilitates effective management and secures healthcare data storage within a digital archive, with access limited exclusively to authorized personnel. The most effective data management and access control approach involves recording all transaction information on the blockchain. Due to constraints related to storage capacity, scalability, and cost, medical data is stored in cloud environments. Saberi et al. [51] developed a break-glass model that integrates blockchain, ABAC, and IPFS to ensure the security and confidentiality of patient records. This healthcare model provides timely access to EMRs and EHRs. Blockchain records all access requests and manages the granting and revocation of access to medical records. The Break-Glass mechanism enables each enrolled EMR owner to grant permission for health professionals to access their medical data. Authorized parties utilize healthcare permissions in emergencies to establish new attributes on electronic medical records (EMRs).

ACTION-EHR [52] focused on enhancing the mobility and accessibility of healthcare data to enable secure and reliable data sharing and management. This system, which is centered

on the patient, allows the patient to manage health data. This system enables the secure aggregation and dissemination of medical data. The health data generated is encrypted using a hybrid cryptosystem that incorporates both public key and symmetric-key cryptography, with the encrypted data subsequently stored in the cloud. Data sharing and information exchange present significant challenges in EHRs. Chelladurai et al. [53] proposed a smart health system for the exchange of medical data, employing a modified Merkle tree to ensure an immutable log while facilitating data accessibility, record updates, data exchange, and viewership on the blockchain. Healthcare data is recognized as susceptible to various attacks, with a significant rise in incidents of hacking targeting this information.

BIoTHR [54] is an IoT-based private blockchain-assisted EHR system that emphasizes privacy preservation and secure data exchange by implementing blockchain technology and swarm exchange techniques. This swarm exchange paradigm enhances the BIoTHR by incorporating agility, flexibility, transparency, high availability, and security. The primary rationale for integrating blockchain into healthcare is to improve performance, establish access control, ensure data security, and facilitate secure data exchange. The supply chain for pharmaceutical products and drugs necessitates adherence to safety, security, data provenance, and governance standards within the healthcare system. The medical supply chain has encountered traceability, transparency, shipment expiration, and tracking challenges. The proliferation of medical data requires a diverse array of medical products and pharmaceuticals.

Detecting counterfeit drugs is complex due to their similarity to the original products. Blockchain facilitates access control for each participant through smart contracts, thereby ensuring traceability, visibility, and safeguarding the activities of individual participants [55]. Monitoring the journey of medication from production to the consumer, along with documenting any adverse effects for future reference, aids the pharmaceutical industry in reducing drug counterfeiting [56]. The Medledger system, utilizing HLF blockchain and chaincode, facilitates drug traceability to combat counterfeit drugs [57]. The solution utilizes decentralized file storage to manage private and sensitive data, ensuring traceability and transparency. To mitigate the proliferation of counterfeit goods, PharmaCrypt [58], a blockchain-based tool, is introduced. This tool facilitates the rapid scanning of products and time stamps for each item and maintains a comprehensive record of all transactions within the supply chain. This approach enables tracing counterfeit products, thereby ensuring security and safety. During packaging, the package's barcode is scanned, and the hash value of a specific product is recorded on the blockchain. The product's movement within the supply

chain is monitored using tamper-proof timestamps on the blockchain, facilitating thorough traceability and permitting the identification and removal of counterfeit products before consumer access.

BRUINchain [59] is a blockchain-based system that utilizes HLF, a blockchain application server, notification and verification services, and smart contracts to facilitate drug tracing, tracking, and verification. BRUINchain scans the barcodes of drug packages, identifies expired products, detects suspected and expired items, quarantines counterfeit drugs, and verifies products directly with the manufacturer. Blockchain aims to transform the healthcare sector into a smart system by offering decentralization, immutability, security, and transparency. This technology facilitates secure access and exchange of EHR, EMR, and PHI, enhances the efficiency of pharmaceutical supply chains, ensures secure storage of medical insurance, and improves hospital waste management. Pham et al. [60] developed a remote healthcare system for doctors, patients, and hospitals that selectively captures essential medical data and stores it on a blockchain, thereby minimizing its size. MIStore, [61], is a medical insurance storage system that utilizes blockchain technology to enhance user credibility. MIStore may aid the insurance company in obtaining information related to patients' healthcare expenditures.

2.2.3.5 How does blockchain handle the security issues of the healthcare system?

Eavesdropping on patients' vital information represents a significant security threat within the healthcare industry. The information may include the patient's ID, address, location, and medical condition. Blockchain employs cryptography to ensure patient protection and anonymity, verify claims made by participating entities regarding the asset, and authenticate the data. Blockchain uses encryption algorithms for data added to the system, rendering this encrypted information difficult for attackers to decipher. Blockchain technology addresses the challenge of securing patient information and the identities of healthcare providers.

SEMRES [62] is a blockchain-based medical framework emphasizing secure data exchange via a triple encryption authentication architecture. The framework comprises three modules: combined encryption and decryption architecture (CEDA), decentralized EMR repository, and content verification. CEDA employs a hybrid mechanism of AES and RSA cryptographic algorithms to encrypt medical data, resulting in a secure encrypted medical record. A decentralized EMR repository employs SHA256 to produce the secure encrypted medical record (SMeR) hash value. The data owner employs their private key to sign the block's

content for decryption and secure data exchange. This content block includes the index of SMeR within the Decentralized EMR repository (DERy), the hash of SMeR, and the authority status. This signed block demonstrates the data owner's authorization and may serve as the entry point for the Decentralized EMR repository. Content decryption and verification occur post-data transmission to confirm the accuracy of the data.

A breach in the healthcare system may result in the unauthorized disclosure of sensitive and confidential medical information. Centralized data storage attracts malicious users and increases the risk of cyber attacks. HealthBlock [63] is a blockchain-based healthcare data management system designed to improve the privacy and security of EHRs through decentralized data storage. HealthBlock offers protection against security vulnerabilities, including spoofing attacks, tampering threats, and repudiation, by implementing fabric certificates, cryptographic functions, and fabricated digital signatures for designated tasks. Regular patient monitoring is essential in healthcare for managing severe health issues and chronic diseases.

BlockMedCare [64] is a blockchain-based healthcare system that integrates IoT and IPFS, focusing on continuous remote patient monitoring. BlockMedCare emphasizes security, scalability, and processing efficiency, ensuring security through implementing blockchain technology and a proxy re-encryption cryptosystem. In encryption techniques, both the public key and secret key are obtained by the patient ($pk_{patient}$, $sk_{patient}$) and the physician ($pk_{physician}$, $sk_{physician}$). The patient encrypts the data with $pk_{patient}$ and generates Prek through pre.rekey($sk_{patient}$, $pk_{physician}$). Upon receiving the encrypted data, Prek Hospital re-encrypts it and transmits it to the physician, who then decrypts it using their private key.

The healthcare sector is susceptible to numerous attacks, rendering protected and encrypted data vulnerable. An attacker within the blockchain can manipulate patient data; however, this requires substantial computational power, extensive resources, and significant energy expenditure. A 51% attack undermines the integrity of the blockchain network, potentially revealing critical details and the data storage location in the cloud. Nonetheless, because the suitable ABE secures the data encryption key, an attacker is unable to access medical information unless the intricate issue of Decisional Modified Bilinear Diffie–Hellman (DMBDH) is resolved [65].

As introduced by Sahai et al. [66], ABE has become the prevailing framework in cryptographic access control for encrypted data. Bethencourt et al. [67] contributed significantly

to CP-ABE by proposing a tree-based access structure scheme. This scheme demonstrates greater adaptability and practicality than KP-ABE. Although ABE, as suggested by Bethencourt et al. [67], Waters, B. [68], Sahai et al. [66], and related systems, provides the benefits of one-to-many encryption, it still faces significant security concerns. The designated authority responsible for key issuance to all eligible users can decrypt any ciphertext using the distributed secret keys. Riepel et al. [69] implement FABEO, a CP-ABE scheme characterized by more concise ciphertexts, reduced pairings, and improved efficiency in optimal and adaptive security contexts.

The authors [70] propose a verifiable encryption scheme that utilizes blockchain technology to improve security and accountability in data management. The multi-authorization mechanism of this scheme allows different authorities to manage user attributes, enhancing flexibility and scalability in access control. The performance analysis demonstrates the scheme's effectiveness; however, empirical evidence is necessary to confirm its efficiency and overall impact.

Thakur et al. [20] present a secure access control mechanism that employs blockchain and ABE. The proposed method utilizes IPFS to store encrypted data, addressing the limitations of blockchain resources and the reliability issues associated with centralized storage systems. This method enhances security and adaptability while facilitating attribute policy hiding and revocation.

2.2.3.6 What are the benefits and challenges of incorporating blockchain in healthcare?

Among numerous blockchain applications, healthcare represents a critical area where blockchain can effectively demonstrate its capabilities [71], [72]. Disseminating data and ensuring access to medical information for verified healthcare providers, researchers, patients, and other stakeholders is essential for enhancing the quality of the healthcare system. The application of blockchain technology in healthcare is expanding rapidly, facilitating clinical data sharing, preserving medical histories, enabling global data exchange, supporting medical research, controlling access to healthcare data, managing drug supply chains, and streamlining clinical trials [73].

The research areas in the healthcare domain include remote patient monitoring, pharmaceutical supply chains, clinical trials, EHRs, and health insurance claims. The decentralized approach of blockchain effectively addresses the challenges present in traditional systems.

SCACIoMT utilizes blockchain technology to protect patients from malicious activities and vulnerabilities [74].

The primary goal for implementing blockchain technology in healthcare, as well as other sectors, is to enhance the security of the infrastructure and protect against unauthorized or malicious entities [5]. The blockchain offers numerous advantages, such as secure data exchange and regulatory sharing through smart contracts, management synchronization of health systems, access control, and secure data storage [75]. The implementation of blockchain technology offers numerous **advantages**, including enhanced security and privacy for health records, protection of patient identity through anonymity, consistent remote monitoring of patient data, and improved accessibility of healthcare information.

[1] **Secure EHR:** Blockchain technology enhances healthcare data storage, management, and interchange by protecting privacy and security. A blockchain smart contract builds and updates immutable health records, enables secure data exchange, and unifies scattered data [76].

Malicious entities can sabotage records, so pairing-based cryptography is used to prevent manipulation, tampering, and forging attacks [77]. Blockchain-based EHR allows healthcare agents and other entities to update patient data in real-time and gives patients more control over their medical information.

[2] **Improved remote patient monitoring:** A blockchain-based RPM improves healthcare quality when patients who need urgent treatment cannot visit the hospital. Real-time data is crucial and challenging in healthcare [78]. The patient has medical or wearable gadgets that measure water, blood pressure, oxygen, and more. The smart contract analyses collected data and initiates alerts based on threshold values [79]. To monitor patients remotely, a smart gadget delivers blood pressure data to HealthContractCaller and BloodPressureMonitor contracts [80]. The patient, doctor, and health center are notified if blood pressure exceeds limits.

[3] **Secure health information exchange:** In healthcare, data sharing and interoperability are the main concerns to ensure that only authorized entities can access medical records, that all system components communicate properly, and that there is no data inconsistency between remote data providers and recipients. FHIRChain [81] houses the HL7 Fast Healthcare Interoperability Resources (FHIR) paradigm for scalable healthcare data sharing, meeting ONC criteria. EdgeMediChain [82] combines

edge computing and blockchain for secure and scalable health data exchange. Address privacy and data inconsistency. In HIE, Zhuang et al. [83] developed a patient-centric data exchange system using smart contracts, DLT, and blockchain.

[4] **Improved drug traceability and pharmaceutical supply chain:** Pharmaceutical drug supply chain management is vital in medicine. However, counterfeiters cause many problems and losses. Counterfeit pharmaceuticals are unapproved and of low quality. Blockchain can ensure the legitimacy and standardization of medicine from manufacture to consumption [73]. Liu et al. [84] proposed an intelligent approach for medicine traceability. Five layers make up the BIoT platform architecture. To ensure secure drug tracing and monitoring, our blockchain architecture strictly regulates drug management, from data collection to device identity and transaction verification to drug quality feedback.

[5] **Clinical trials:** Data transparency may inhibit selective positive reporting. Sometimes, the clinical trial protocol must be kept private from stakeholders because it contains scientific and commercial secrets. Blockchain might provide real-time trial data recording. Every trial transaction is timestamped and kept in a blockchain, allowing stakeholders to transfer consent in real-time. Blockchain technology aids in patient admission, data management, and analysis in clinical trials [85]. Albanese et al. [86] introduced a Hyperledger blockchain-based dynamic consent management strategy (SCoDES) in clinical trials to enhance patient consent monitoring and data privacy. Zhuang et al. [87] updated the CTMS by introducing Quorum characteristics, resulting in a safe, auditable, and universal system. Healthcare benefits from blockchain include immutability, decentralization, security, transparency, and traceability. Identity management, scalability, interoperability, cost unpredictability, and legal and regulatory difficulties remain. These hurdles impede users and stakeholders from globalizing this technology.

Blockchain solves several security issues. Blockchain-based system development apps may inherit or directly exhibit these risks. These issues are especially concerning in healthcare, as vital data may be compromised. The main goal of this RQ is to address implementation **problems** in blockchain healthcare solutions. The researcher seeks to overcome the challenges of incorporating blockchain in healthcare solutions.

[1] **Challenges related to data security:** Blockchain presents various security chal-

lenges attributable to its architecture and execution. Blockchain addresses weaknesses, including double-spending attacks, anonymity concerns, difficulty rising attacks, block withholding attacks, consensus-based attacks, selfish mining attacks, mining malware, timejack attacks, Finney attacks, race attacks, 51% attacks, and block discarding attacks. These attacks exploit consensus mechanisms, peer-to-peer networks, and flaws in smart contracts. The resolutions to these problems can be addressed by overseeing node behavior, enhancing consensus mechanisms, and augmenting hash rate [88].

[2] Privacy leakage: The primary necessity of the healthcare sector is to ensure the anonymity and privacy of patients' health information and identities. Healthchain [89] emphasizes privacy protection by integrating a permissioned blockchain and utilizing off-chain storage via IPFS for EHRs. It also incorporates a cryptographic public key encryption mechanism to ensure data integrity, privacy, and security. The hybrid privacy-preserving framework, hOCBS [90], utilizes a consensus mechanism and off-chain storage to provide secure data sharing, privacy preservation, and dynamic consent management. Additional critical technologies and cryptographic solutions that constitute essential components for safeguarding privacy in blockchain include secure multi-party computation, homomorphic encryption, ring signatures, and zero-knowledge proofs, among others [16].

[3] Scalability: Private clinics, healthcare facilities, insurance companies, and patients generate data via wearable devices. Due to the time and computer resources required to handle large amounts of data, latency and blockchain performance suffer. As nodes increase, the blockchain lacks scalability.

The study [91] proposes IPFS off-chain storage to overcome scalability issues in blockchain-based healthcare systems. IPFS storage receives encrypted documents from the patient's public key. While solving scalability issues, IPFS protects medical data. Allowing only allowed nodes access to data lets the user control access. BCHealth improved its privacy-preserving architecture's performance and scalability by optimizing its blockchain network and modifying its Proof of Authority (PoA) consensus algorithm [92]. The data owner controls access. To ensure scalability, BlockMedCare [64] uses IPFS for off-chain database storage.

[4] Interoperability: No healthcare-specific blockchain standard exists to facilitate cross-platform application communication. Platforms may provide difficulties be-

cause of intricate smart contract functionalities, consensus protocols, and transaction methodologies [93]. Interoperability enables two blockchains to collaborate, yet no standardized protocol facilitates integration. The absence of blockchain interoperability hinders seamless information transfer, restricting access to collaborative patient data for a comprehensive view and critical healthcare information. A potential solution is to standardize open protocols for intercommunication among blockchain platforms [51], [53]. Cross-chain interoperability enables communication between blockchain systems, whether homogeneous or heterogeneous. Sidechain, blockchain routers, and smart contracts are proposed solutions for interoperability [75], [93].

[5] **Technical complexity and Issues:** Implementing blockchain in healthcare is complicated. Massive data sets can delay the system and cost more to store. Adding blockchain to healthcare infrastructures is similarly complex. To successfully integrate blockchain into healthcare systems, considerable changes are needed, including financing, planning, knowledge, and time [94]. Poor data standardization limits healthcare providers from sharing and analyzing patient data effectively [95]. Technology and network infrastructure are needed for blockchain in healthcare. A compute node network validates transactions and produces blocks in this distributed ledger system. Concerns about unexpected operational costs persist for long-term maintenance [96].

[6] **Difficult to use:** Blockchain integration into operational frameworks presents significant challenges due to its complexity. The present complexity of blockchain necessitates considerable technical proficiency, encompassing the management of public/private cryptographic keys and compliance with designated procedural standards, thereby limiting its user-friendliness and accessibility.

[7] **Computing Power Consumption:** Decentralized consensus mechanisms utilizing verification-centric algorithms in blockchain guarantee transaction consistency and validity. Public blockchains extensively employ consensus methods, resulting in considerable energy consumption [97]. Individual healthcare institutions independently manage their data in systems such as Medrec [49]. The security and accessibility of data depend on the stewards of that data. The system employs the Proof of Work mechanism to attain consensus, notwithstanding the limitations of restricted processing speeds, protracted transaction confirmations, and elevated energy usage.

[8] **Cost:** Blockchain in healthcare raises cost concerns. Software and hardware modifications increase implementation costs. Investments in operations include maintenance,

monitoring, upgrading, cybersecurity, and regulatory compliance [98]. Blockchain training for healthcare staff costs more. Financial investment is needed to integrate data with existing systems and enhance blockchain solutions to accommodate growing data quantities [99]. Security needs encryption, access controls, and compliance for patient data. Effective cost management in healthcare requires strategic decision-making, stakeholder participation, and understanding of the technology's potential and financial impact [100].

[9] **Latency:** A decentralized blockchain assures data confidentiality and consistency, but confirming and recording transactions takes time and computer effort. Medical care efficacy depends on addressing latency issues. To reduce latency and improve blockchain-based healthcare systems, optimize blockchain protocols, use hybrid models, and improve network infrastructure.

[10] **Regulatory Concerns:** Incorporating blockchain into healthcare encounters a substantial obstacle: guaranteeing adherence to current standards and data privacy legislation. Collaboration between healthcare stakeholders and regulatory bodies is essential for widespread use [71], [57]. A primary difficulty is the scalability of blockchain systems across many locations, where regulatory frameworks may be inconsistent. Government restrictions and cultural intricacies present significant obstacles to adoption. Comprehensive stakeholder engagement and endorsement are essential to mitigate these uncertainties and secure acceptance from the global healthcare community [96], [95], [5].

2.3 Research Gaps

Based on the literature review, several research gaps have been identified:

[1] There are few studies that investigate the practical performance characteristics of blockchain networks under varying conditions.

[2] While blockchain applications in EHR, EMR, and PHI are well-studied, areas such as patient autonomy, clinical trials, pharmaceutical supply chains, and remote patient monitoring are comparatively less explored. Existing studies in these domains are limited in scope and depth.

[3] Despite its advantages, blockchain technology faces significant security and privacy challenges, highlighting the need for additional security measures.

[4] Existing studies are using smart contracts as the access control, and since there are numerous vulnerabilities in the smart contract, adopting the access control mechanism is a pressing requirement.

[5] It is not feasible to store the data in on-chain storage, and incorporating the private blockchain is not enough to make the system scalable.

2.4 Performance Evaluation Metrics

The following performance evaluation metrics are employed in this thesis to provide a comprehensive assessment of system performance:

[1] **Throughput :**Throughput shows how many transactions the network processes in a given time. Throughput performance measures blockchain network transaction processing speed and efficiency. Read throughput measures performance by counting successful read operations over time. Reads per second count the number of read operations done in the interval.

$$\text{Read Throughput} = \frac{\text{Total read operations}}{\text{Total time in seconds}} \tag{2.4.1}$$

Transaction throughput (TPS) is the rate at which valid transactions are confirmed and added to the blockchain in a given time frame. This performance statistic considers all nodes to assess network efficiency.

$$\text{Transaction Throughput} = \frac{\text{Total valid committed transactions}}{\text{Total time in seconds}} \tag{2.4.2}$$

[2] **Latency :** The latency of a transaction is calculated by finding the difference between the time taken to complete the transaction and the time taken to deploy it. Read latency is the time between the response to the request and the submission of the read request.

$$\text{Read Latency} = \text{Reply received time} - \text{Request submission time} \tag{2.4.3}$$

On the other hand, the amount of time needed for a transaction to become available throughout the entire network is stated as transaction latency. This period encompasses when the transaction is first submitted and becomes widely accessible on the

network.

$$T_{Latency} = \text{Confirmation time} \times \text{Network threshold} - \text{Transaction submission time} \tag{2.4.4}$$

[3] Success Rate : The success rate of a blockchain can be measured by evaluating the ratio of successfully executed transactions to the total number of transactions attempted. Successful transactions refer to those completed without encountering any errors or exceptions during their execution.

[4] Resource Consumption : During the test execution, transactions utilize CPU, cache memory, disk, and RAM resources. Extreme consumption of these resources, e.g., disks, can significantly impact the system's performance. The Resource Monitor allows initiation/termination of monitoring and obtaining information about the resource usage of the underlying blockchain system, such as network I/O, memory, CPU, and so on. The Docker resource monitor provides access to several performance metrics, including maximum and average memory usage, maximum and average CPU usage as a percentage, inbound and outbound network traffic, disk read operations, and disk write operations. In network I/O, traffic in and out refers to the amount of data received by a peer or orderer node (traffic in) and the amount of data sent out from the node to other network participants (traffic out).

[5] Agreement Time : It refers to the time it takes for a consensus mechanism to agree on the validity of a transaction or block within the network. Consensus is the process through which all nodes in the blockchain network reach an agreement on the state of the blockchain, ensuring that they all have the same data.

[6] Computation Time : It refers to the time required to perform encryption, decryption, and key generation operations in the cryptographic process.

[7] Accuracy : This metric indicates the overall correctness of predictions of the model. It is calculated as the ratio of correctly predicted instances (true positives and negatives) to the total number of instances.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} \tag{2.4.5}$$

[8] Precision : Precision focuses on the accuracy of positive predictions. It answers the

question, "Of all the items labeled as positive, how many were actually positive?"

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.4.6}$$

[9] Recall : Recall, also known as sensitivity, assesses the model's ability to identify all relevant instances. It addresses the question: "Of all the actual positive instances, how many did the model correctly identify?"

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.4.7}$$

[10] F1-Score : F1 score serves as a harmonic mean of precision and recall, providing a balanced measure that is particularly useful when there is a trade-off between these two metrics. It is calculated using the following formula:

$$F1 - Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.4.8}$$

[11] Micro precision : Micro Precision and Micro Recall aggregate the contributions of all classes to compute the average metric. This approach treats each instance equally, regardless of its class.

[12] Micro Precision is calculated as:

$$\text{Micro Precision} = \frac{\sum \text{True Positives}}{\sum \text{True Positives} + \sum \text{False Positives}} \tag{2.4.9}$$

[13] Micro Recall is calculated as:

$$\text{Micro Recall} = \frac{\sum \text{True Positives}}{\sum \text{True Positives} + \sum \text{False Negatives}} \tag{2.4.10}$$

[14] Micro F1 Score: It is a performance metric used in multi-class classification problems that combines the concepts of precision and recall into a single score. It is particularly useful when dealing with imbalanced datasets, as it treats all instances equally, regardless of their class.

$$\text{Micro F1-Score} = 2 \times \frac{\text{Micro Precision} \times \text{Micro Recall}}{\text{Micro Precision} + \text{Micro Recall}} \tag{2.4.11}$$

[15] Hamming Loss : Hamming loss measures the prediction error by considering both

the instances where an incorrect label is assigned and the cases where a relevant label is omitted.

$$L_{hamming}(d, \widehat{d}) = \frac{1}{S_n L} \sum_{t=0}^{S_n-1} \sum_{u=0}^{L-1} 1(d_{t,u} \neq \widehat{d}_{t,u}) \tag{2.4.12}$$

in the equation $\widehat{d}_{t,u}$ = predicted value for u^{th} label

t - given label

$d_{t,u}$ = corresponding true value

S_n = number of samples

L = number of labels

A lower hamming loss indicates the better performance of a model for an ideal classifier.

[16] Jaccard Similarity : In multi-label classification, the Jaccard similarity (or Jaccard index) quantifies the similarity between two label sets by calculating the ratio of the size of their intersection to the size of their union.

$$J(tl, pl = \frac{|tl \bigcap pl|)}{|tl \bigcup pl|)} \tag{2.4.13}$$

tl = a ground truth label set

pl = a predicted label set.

In the next chapter (Chapter 3), we provide an introduction to blockchain technology and its fundamental technical components. The chapter also outlines the various applications of blockchain across different domains and presents a brief overview of the key preliminaries relevant to this thesis.

Chapter 3

Preliminaries

The purpose of this chapter * *is to provide brief information about the several basic concept of blockchain. This chapter examines the applications of blockchain and its potential to transform various industries. This chapter discusses the consensus protocol, smart contract, and ABE algorithm to establish a foundational understanding and familiarity with the research work.*

3.1 Blockchain

In 2017, Seebacher et al. [101] defined blockchain as a decentralized database collaboratively maintained and validated by a peer-to-peer network. It consists of a sequence of interlinked blocks, each carrying time-stamped transactions. These transactions are safeguarded by public-key cryptography and are authenticated by the network's community. The fundamental architecture of blockchain functions on a peer-to-peer paradigm, distinguishing it from server-based or centralized systems, wherein all participating nodes engage directly without the mediation of a central authority [102]. Blockchain fundamentally constitutes a

*The part research work covered in this chapter has been published/accepted/communicated in: Anita Thakur, Virender Ranga and Ritu Agarwal, "Exploring the Transformative Impact of Blockchain Technology on Healthcare: Security, Challenges, Benefits, and Future Outlook," *Transactions on Emerging Telecommunications Technologies* , Wiley, vol. 36, p. e70087, 2025, DOI: https://doi.org/10.1002/ett.70087. **(SCIE, IF=2.5)**
&
Anita Thakur, Virender Ranga and Ritu Agarwal, "Workload dynamics implications in permissioned blockchain scalability and performance," *Cluster Computing*, Springer, vol. 27, issue. 8, pp.11569-11593, 2024, DOI: https://doi.org/10.1007/s10586-024-04550-z. **(SCIE, IF=3.6)**
&
Anita Thakur, Virender Ranga and Ritu Agarwal, "Revocable and Privacy-Preserving CP-ABE Scheme for Secure mHealth Data Access in Blockchain," *Concurrency and Computation: Practice and Experience*, Wiley, vol. 37, p. e70064, 2025, DOI: https://doi.org/10.1002/cpe.70064. **(SCIE, IF=1.5)**
&
Anita Thakur, Virender Ranga and Ritu Agarwal, "ABHealChain: Enhancing Privacy and Security in Healthcare Data Sharing through Hyperledger Fabric and Attribute-based Access Control", —communicated

distributed public ledger data structure that facilitates decentralized storage and organization of data records. Each participant in the network upholds a consolidated record of every transaction. The transactions are structured into interconnected blocks, guaranteeing the integrity and accuracy of the data throughout time. Blockchain fundamentally constitutes a sequential configuration of immutable records, with each legitimate block cryptographically connected to its preceding valid block (as illustrated in Figure 3.1).

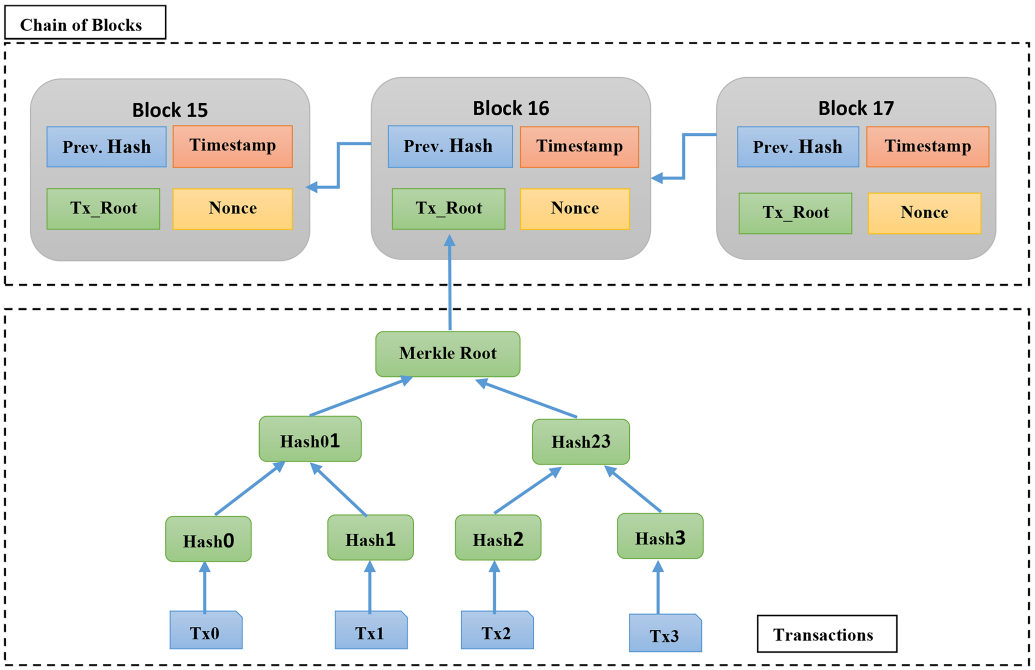


Figure 3.1: Blockchain structure with interconnected blocks.

A blockchain is composed of a sequence of blocks, each containing a header and a body. The block header holds essential information such as the hash of the previous block (H_{prev}), a timestamp (T), and a nonce (N), while the block body is dedicated to storing transaction data (D). The relationship among these components can be expressed mathematically as: $[H = Hash(H_{prev}||T||N||D)]$. This structure is crucial for maintaining data integrity and establishing a chronological sequence within the blockchain. Each block is interconnected with its predecessor through the hash of the previous block, forming a secure and immutable chain.

3.1.1 Technical Background

To adequately understand the functioning of blockchain, one must comprehend the essential components and concepts that support it (as depicted in Figure 3.2).

[1] **Smart Contract**: A smart contract is a self-executing agreement encoded in lines of code that facilitates transactions between parties. It constitutes a contract between the purchaser and the vendor. It is a decentralized contract encoded in a computer language and documented on the Blockchain. A smart contract was initially proposed by Nick Szabo in 1997 [103], who created the cryptocurrency "Bit Gold" in 1998 and characterized smart contracts as a computerized transaction system that enforces the contract conditions. Nick Szabo defined a smart contract as an automated transaction protocol that enforces the conditions of a contract. The primary aims of smart contract design are to fulfill standard contractual requirements (including payment periods, liens, secrecy, and enforcement), reduce deliberate and inadvertent exceptions, and decrease reliance on trusted intermediaries. Associated economic objectives encompass the reduction of fraud losses, arbitration and enforcement expenses, and various transaction costs. Smart contracts are authored in a high-level programming language and compiled into Ethereum Virtual Machine bytecode. Solidity is a computer language renowned for developing smart contracts. Each smart contract is located at a specific address on the Blockchain. Smart contracts engage with one another using function calls that encompass function names and parameters. To execute the contract's code, a function call is binary encoded and transmitted to the contracts under the transaction's data field [104].

[2] **Merkle Tree**: A Merkle tree, also known as a binary hash tree, is a data structure used in computer science and cryptography to efficiently and securely verify the integrity of large sets of data. It is particularly useful in blockchain and peer-to-peer networks [105]. The structure consists of leaf nodes representing individual data blocks and non-leaf nodes comprising hashes of their respective child nodes. In a Merkle tree, each leaf node contains the hash of a data block, while each non-leaf node contains the hash of the concatenation of its two child nodes. It can be mathematically represented as: $[H_i = Hash(H_{left}||H_{right})]$, where H_i is the hash of the parent node, $H_{left}$is the hash of the left child, and H_{right} is the hash of the right child. The process continues recursively until a single hash, known as the Merkle root, is obtained at the top of the tree. This root hash serves as a compact representation of all the data in the tree.

[3] **Cryptographic Hash Function** A cryptographic hash function is a fundamental component of blockchain, serving multiple critical roles in ensuring the security, integrity, and functionality of the blockchain. A cryptographic hash function takes

an input (or "message") and produces a fixed-size string of characters, typically a sequence of numbers and letters. This output is known as the hash value or hash digest. Consider the equation: $\mathfrak{y} = F(\mathfrak{x})$, where $\mathfrak{x}$ represents a string of any length and $F()$ is a one-way hash function, with $\mathfrak{y}$ as the resulting output. One-way hash functions are designed such that, while it's possible to compute $\mathfrak{y}$ from $\mathfrak{x}$ and $F(\mathfrak{x})$, there should be no efficient method or deterministic algorithm that can reverse the process to recover $\mathfrak{x}$ from $\mathfrak{y}$ and $F(\mathfrak{x})$.

[4] Consensus Algorithm Blockchain consensus encompasses the various mechanisms and protocols that enable participants in a blockchain network to agree on the state of the distributed ledger. This process is essential for ensuring the integrity, security, and reliability of the blockchain, particularly in a decentralized setting where no single entity governs the entire network. Different consensus mechanisms have been developed, each with unique advantages and drawbacks [4].

In a blockchain network, miners (or validators in some systems) work together to validate new blocks of transactions. This consensus process ensures that all participants agree on the ledger's state and prevents fraudulent activity. If a block fails this validation, it is promptly discarded, as depicted in Figure 3.3. In a blockchain network, the prevailing principle dictates that the longest chain is considered legitimate. The probability denoted as Q_z quantifies the likelihood of the attacker node outpacing the honest chain by z blocks [106], [107], [108].

$$Q_0 = 1, Lim_{z\to\infty} \quad Q_z = 0 \tag{3.1.1}$$

$$Q_z = \mathfrak{p}Q_{z+1} + \mathfrak{q}Q_{z-1}, z = 1, 2, 3, ..., \infty \tag{3.1.2}$$

if $\mathfrak{p} > \mathfrak{q}$,

$$A_z = Q_{z+1} - Q_z, \mathfrak{r} = \mathfrak{q}/\mathfrak{p} \tag{3.1.3}$$

from the equation Equation 3.1.2, we get

$$Q_z - Q_0 = \sum_{\mathfrak{k}=0}^{z-1}(Q_{\mathfrak{k}+1} - Q_{\mathfrak{k}}) = (1 - \mathfrak{r}^z/1 - \mathfrak{r})A_0 \tag{3.1.4}$$

from the equation Equation 3.1.1 we get

$$Q_z = \mathfrak{r}^z = (\mathfrak{q}/\mathfrak{p})^z, \mathfrak{p} > \mathfrak{q} \tag{3.1.5}$$

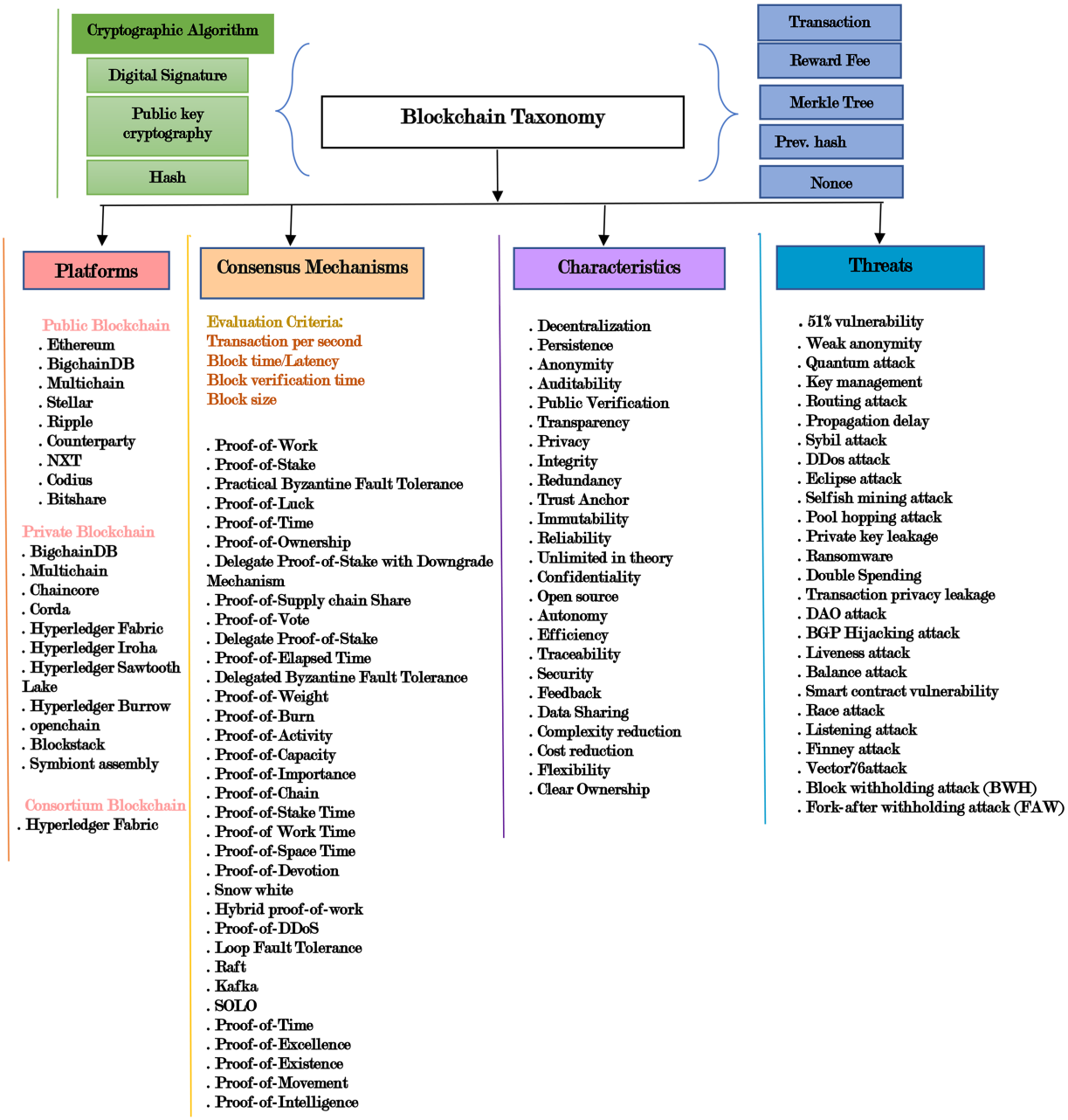


Figure 3.2: Taxonomy of blockchain showcasing platform, consensus mechanism, characteristics, threats

$$Q_z = 1, \mathfrak{p} \leq \mathfrak{q} \tag{3.1.6}$$

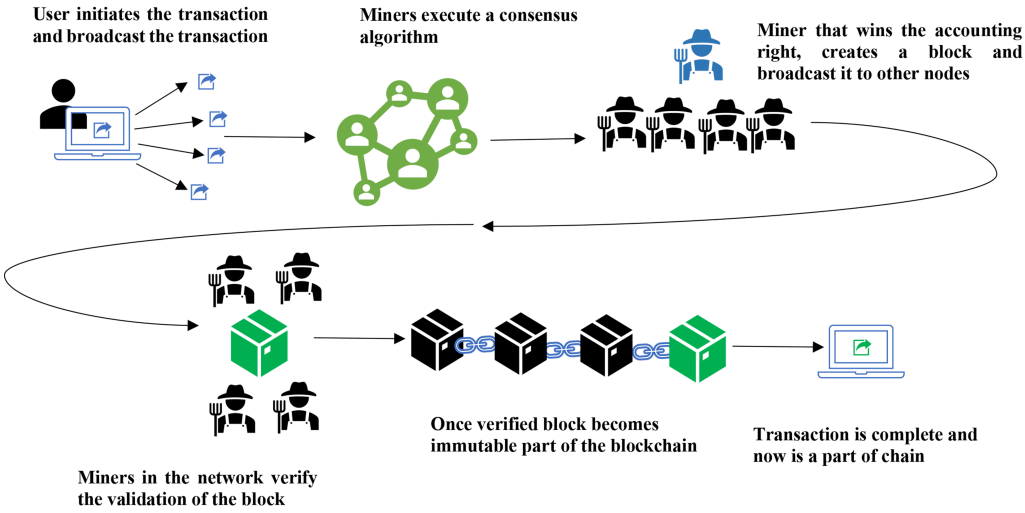


Figure 3.3: Workflow of blockchain

we can say that

$$Q_z = \begin{cases} 1, & \mathfrak{p} \leq \mathfrak{q} \\ (\mathfrak{q}/\mathfrak{p})^z, & \mathfrak{p} > \mathfrak{q} \end{cases} \tag{3.1.7}$$

where $\mathfrak{p}$ = probability of legitimate node mining the block

$\mathfrak{q}$ = probability of the attacker mining the new block before the legitimate node

Q_z= probability that attackers catch up with honest nodes from z blocks

For a blockchain network to remain operational, its participants must reach a consensus about the current state of the shared ledger and how data is packaged into blocks. In a distributed system, a consensus protocol ensures that all participating nodes agree on the same state of the system, particularly the order in which transactions occur. The consensus mechanism allows reading and updating to a certain state that guarantees the transaction's ordering and the integrity of the content across geographically distributed areas in a decentralized manner. The topology of the network agents is depicted using a directed graph $G = (V, E)$ by the formulas given below [109], [110].

V represents the Nodes $V = 1, 2, 3...n$ and E represents the Edges $E \subseteq V \times V$,
$N_i = j \in V : (i, j) \in E$.

Neighbors of agents i is represented by N_i. To solve the consensus problem and reach an agreement regarding the state of n agents with dynamics

$$\dot{x}_i(t) = u_i(t) \tag{3.1.8}$$

can be represented as an n^{th} order linear system:

$$\dot{x}_i(t) = \sum_{j \in (N_i)} (x_j(t) - x_i(t)) + b_i(t), \quad x_i(0) = z_i \in \mathbb{R}, b_i(t) = 0 \tag{3.1.9}$$

The collective dynamics of the following group of agents Equation 3.1.9 can be written as follows:

$$\dot{x} = -Lx \tag{3.1.10}$$

L is graph Laplacian of network $L = [l_{ij}]$

$$Q_z = \begin{cases} -1, & j \in N_i \\ |N_i|, & j = i \end{cases}$$

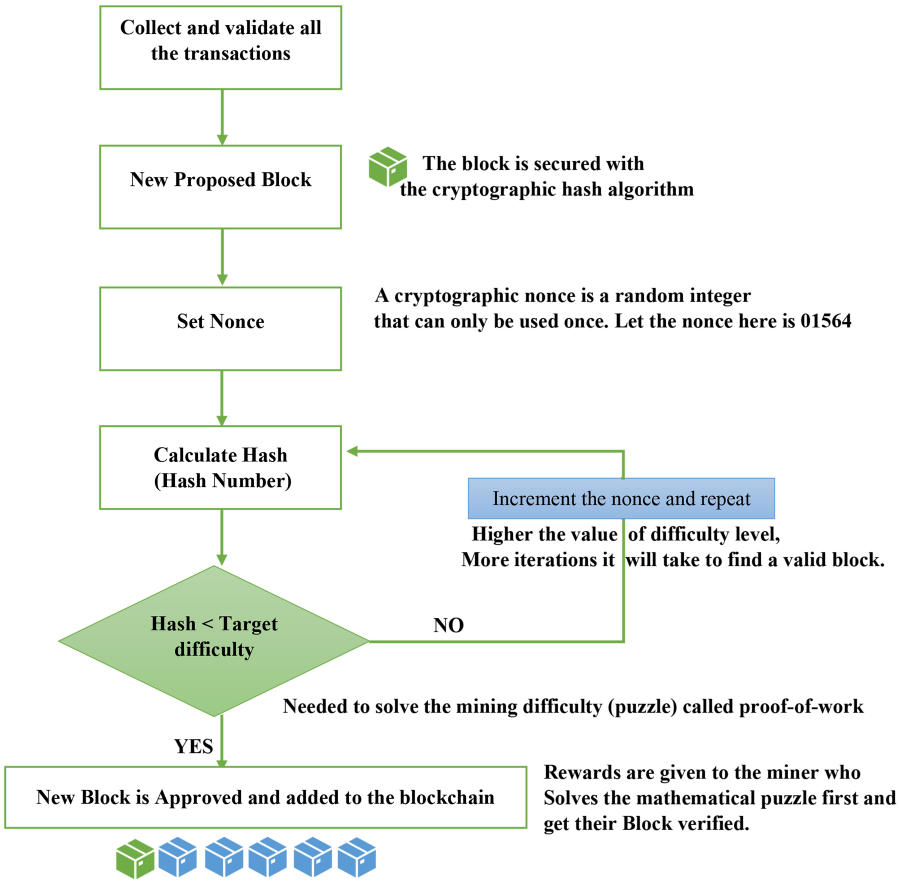


Figure 3.4: Flow of PoW consensus mechanism

Various blockchains employ distinct consensus mechanisms, including PoS [111], PoW [112], PBFT [113], and PoA [114]. PoW is a consensus process delineating the validity verification for participants or miners. In the PoW process illustrated in Figure 3.4, all transactions are aggregated and consolidated to create a block. A challenge or difficulty is established for the block, and miners within the network are tasked with solving this intricate mathematical problem. A block is deemed valid only if its reference or computed hash exceeds a specific threshold (Equation 3.1.12). The initial miner to resolve this challenge receives the reward, and the validated block is subsequently appended to the blockchain [115]. The limitation of PoW is its requirement for substantial resources and time to solve the intricate puzzle. In PoW, a majority of malevolent nodes can manipulate the network and execute a 51% attack [116]. The nonce is updated to satisfy equation (Equation 3.1.12), and the hash is iteratively calculated until a valid block is identified.

$$nonce = nonce + 1 \tag{3.1.11}$$

$$Hash(Block) \leq V_t \tag{3.1.12}$$

$$V_t = \frac{\mathcal{M}}{\mathcal{D}} \tag{3.1.13}$$

Hash is a hash function with a variable number range $[0, \mathcal{M}]$. The Target value represented as V_t is a threshold determined by the mining difficulty. The calculated value of the hash has to be less than the threshold value for a proposed block to be included in the chain of blockchain. $\mathcal{D} \in [1, \mathcal{M}]$ is the Target difficulty. Let us find a valid block: a miner with sufficient resource availability and hardware capability performs c operations per second at $c/\mathcal{D}$.

$$\mathcal{P}\{T(c) \leq t\} = 1 - exp(-ct/\mathcal{D}) \tag{3.1.14}$$

There are n miners in the network with the hash rate c_1, c_2,..., c_n.

The time to find the valid block T Equation 3.1.15:

$$\mathcal{P}\{Tdef = min(T_1, T_2, ..., T_n) \leq t\} = 1 - exp(-t\mathcal{D}\sum_{n}^{j=1} c_i) \tag{3.1.15}$$

$$\mathcal{P}\{\mathfrak{T} = T_i\} = c_i / \sum_{n}^{i=1} c_j \tag{3.1.16}$$

To keep the blockchain functioning across the geographically distributed area and to ensure

security, privacy, and correctness, the shared public ledger has consensus mechanisms that are fault-tolerant and maintain the identical blockchain globally.

3.1.2 Popular Blockchain Platforms

Some of the popular blockchain are explained below

[1] **Ethereum**: A decentralized, open-source blockchain platform that enables developers to build and deploy smart contracts and decentralized applications (dApps). It operates on a public blockchain and uses its native cryptocurrency, Ether (ETH), for transactions and computational services.

[2] **Hyperledger Besu:** An open-source Ethereum client designed for enterprise use. It supports both public and private networks and is built to be modular and scalable. Besu allows organizations to leverage Ethereum capabilities while providing features like privacy and permission that are suitable for business applications.

[3] **HLF:** A permissioned blockchain framework under the Hyperledger umbrella, designed for enterprise solutions. It offers a modular architecture, allowing organizations to create private channels and manage data access among participants. Fabric is known for its flexibility, scalability, and strong privacy features.

3.1.3 Applications of Blockchain: An overview

Blockchain, originally created to support cryptocurrencies, has transformed into a multifaceted tool with uses that reach well beyond digital currencies. The decentralized, immutable, and transparent characteristics render it highly appropriate for settings where trust, security, and accountability are critical.

[1] **Finance:** The financial sector has been one of the earliest adopters of blockchain. Cryptocurrencies like Bitcoin and Ethereum leverage blockchain to facilitate peer-to-peer transactions without intermediaries, reducing transaction costs and increasing speed. Additionally, decentralized finance (DeFi) platforms are emerging, allowing users to lend, borrow, and trade assets without traditional financial institutions, thereby democratizing access to financial services.

[2] **Supply Chain Management:** Blockchain enhances supply chain transparency by providing a tamper-proof ledger of transactions. Companies can track the provenance

of goods, ensuring authenticity and ethical sourcing. Retailers are using blockchain to trace food products from farm to table, improving food safety and reducing waste. This level of traceability can also help in recalling defective products more efficiently.

[3] **Healthcare:** In healthcare, blockchain can provide a secure and interoperable platform for managing patient records. By allowing patients to control access to their data, blockchain enhances privacy and security while facilitating data sharing among healthcare providers. This can lead to improved patient outcomes and more personalized care. Additionally, blockchain can combat counterfeit drugs by providing a transparent record of the drug supply chain.

[4] **Real Estate:** Blockchain can streamline property transactions and title management by providing a secure and transparent record of ownership. Smart contracts can automate various processes, such as escrow and title transfers, reducing the need for intermediaries and minimizing fraud. This can lead to faster transactions and lower costs for buyers and sellers.

[5] **Identity Management:** Blockchain offers a decentralized solution for identity verification, reducing the risk of identity theft and fraud. By allowing individuals to control their digital identities, blockchain can streamline processes such as Know Your Customer (KYC) in financial services and enhance security in online transactions. This can also empower individuals in regions with limited access to traditional identification systems.

[6] **Voting Systems:** Blockchain can enhance electoral integrity by providing secure, transparent, and tamper-proof voting processes. By recording votes on a blockchain, stakeholders can ensure that each vote is counted accurately and that the results are verifiable. This application has the potential to increase voter confidence and participation in democratic processes.

[7] **Insurance:** In the insurance industry, blockchain can automate claims processing and underwriting through smart contracts, improving efficiency and reducing fraud. By providing a transparent record of policyholder information and claims history, blockchain can streamline operations and enhance customer trust.

[8] **Intellectual Property:** Blockchain can protect copyrights and royalties by providing immutable records of ownership and usage. Artists and creators can register their work on a blockchain, ensuring that they receive fair compensation for their intellectual

property. This application is particularly relevant in the digital age, where content can be easily copied and distributed.

[9] **Gaming:** In the gaming industry, blockchain enables true ownership of in-game assets through non-fungible tokens (NFTs). Players can buy, sell, and trade these assets on decentralized marketplaces, creating new economic opportunities within gaming ecosystems. This application also allows for greater player engagement and investment in virtual worlds.

[10] **Energy:** Blockchain can facilitate peer-to-peer energy trading, allowing consumers to buy and sell excess energy generated from renewable sources. This decentralized approach can improve grid management and promote the use of sustainable energy solutions. By enabling transparent transactions, blockchain can also enhance trust among energy producers and consumers.

3.2 Hyperledger Caliper

Hyperledger Caliper, a blockchain benchmarking tool, is chosen to assess the performance of the blockchain implementation. One prominent benefit of employing Hyperledger Caliper is its ability to simulate numerous clients capable of injecting workloads into the blockchain network, thereby supporting multiple client threads. CaliperCLI facilitates the execution of benchmarks based on specified settings through a command-line interface [117]. The provided commands are processed by Caliper-Core, which orchestrates the creation of a SUT and ongoing monitoring of its responses. The SUT is seamlessly integrated with each blockchain platform through the Caliper-Adaptor. Hyperledger Caliper currently supports multiple blockchain platforms, including Hyperledger Besu, Ethereum, FISCO BCOS, and HLF [118].

3.3 Bilinear Pairing

Consider two multiplicative cyclic groups $\mathbb{G}_0$ and $\mathbb{G}_T$, both having the same prime order p and g which depicts the generator of $\mathbb{G}_0$. We say $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_T$ is a bilinear pairing if the following properties hold:

[1] *Bilinearity: $\forall a, b \in \mathbb{Z}_\rho$ and $g_1, g_2 \in \mathbb{G}_0$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.*

[2] *Non-degeneracy: There exists $g_1 \in \mathbb{G}_0$, $e(g_1, g_1) \neq 1$*

[3] Computability: $e(g_1, g_2)$ *is efficiently computable for all* $g_1, g_2 \in \mathbb{G}_0$ *in polynomial time.*

3.4 Decisional q-Parallel BDHE Assumption

The definition of the Decisional q-parallel BDHE problem is as described by Waters, B. [68]. Given a bilinear map $(p, \mathbb{G}_1, \mathbb{G}_2, e, g)$*, choose random* $a, s, b_1, ..., b_q \in \mathbb{Z}_p$*. If the adversary* $\mathfrak{A}$ *is given*

$\overrightarrow{y} = g, g^s, g^a, ...g^{a^q}, , g^{a^{q+2}}, ..., g^{a^{2q}}$

$\forall 1 \leq j \leq q, g^{s.b_j}, g^{a/b_j}, ..., g^{a^q/b_j}, , g^{a^{q+2}/b_j}, ..., g^{a^{2q}/b_j}$

$\forall 1 \leq j, k \leq q, \quad k \neq j, g^{a.s.b_k/b_j}, ..., g^{a^q.s.b_k/b_j}$ *it must be difficult to distinguish between* $e(g,g)^{a^{q+1}s} \in \mathbb{G}_2$ *and a random element* $R \in \mathbb{G}_2$*. An algorithm* $\mathfrak{C}$ *guesses* $z \in \{0, 1\}$ *with advantage* ε *in solving the Decisional q-parallel BDHE problem in* $\mathbb{G}_2$ *if*

$$\left| Pr[\mathfrak{C}(\overrightarrow{y}, T = e(g,g)^{a^{q+1}s}) = 0] - Pr[\mathfrak{C}(\overrightarrow{y}, T = R) = 1] \right| \geq \varepsilon$$

As in [68], we can say that the Decisional q-parallel BDHE assumption holds if there is no polynomial time algorithm to solve the Decisional q-parallel BDHE problem with a non-negligible advantage.

3.5 Linear Secret Sharing Scheme

A secret-sharing scheme P over a set of parties P is called linear over $\mathbb{Z}_\rho$ if the following conditions hold:

[1] *The shares for each party form a vector over* $\mathbb{Z}_\rho$*.*

[2] *There exists a share-generating matrix* $M \in \mathbb{Z}_\rho^{l \times n}$ *with* l *rows and* n *columns which can make shares for* $\prod$*. For all* $i \in \{1, ..., l\}$ *the* i^{th} *row of* M *is labeled via a function* ρ*, that associates* M_i *to the party* $\rho(i)$*. Considering the vector* $\overrightarrow{v} = (s, r_2, ..., r_n) \in \mathbb{Z}_\rho^n$*, where* $s \in \mathbb{Z}_\rho$ *is the secret to be shared, and* $r_2, ..., r_n$ *are randomly chosen from* $\mathbb{Z}_\rho$*, then* $M_{\overrightarrow{v}}$ *is the vector of* l *shares of the secret* s *according to* $\prod$*. The share* $M_{\overrightarrow{v}}$ *belongs to party* $\rho(i)$*.*

Waters, B. shows [68] that every LSSS according to the above definition also enjoys the linear reconstruction property, defined as follows: suppose that $\prod$ is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subseteq \{1, ..., l\}$ be defined as $I = \{i :$

$\rho(i) \in S$. There exist constants $w_i \in \mathbb{Z}_\rho$ with $i \in I$ that satisfy that if λ_i are valid shares of any secret s according to $\prod$, then $\Sigma_{i \in l} w_i \lambda_i = s$ holds. According to the access control requirements of the paid data-sharing scenario, our system utilizes LSSS to support any monotonic access structure.

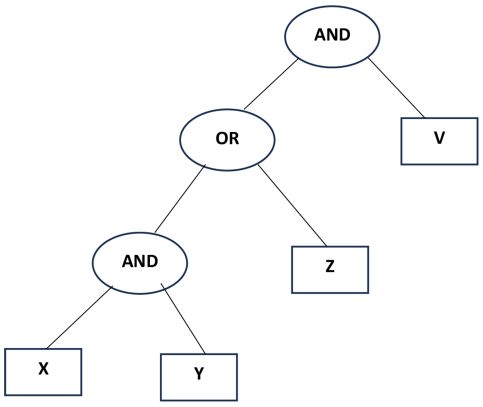


Figure 3.5: Access tree

3.6 Access Structure

Access structures define individuals who possess the requisite qualifications or those who lack permission to access particular data, contingent upon the authorization of specific attribute sets.

Definition: In access structure [119], *Let* $\{P_1, P_2, \ldots, P_n\}$ *be a set of parties. A collection* $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ *is monotone for all* B *and* C*, if* $B \in \mathbb{A}$*,* $B \subseteq C$*, then* $C \in \mathbb{A}$*. An access structure is a collection* $\mathbb{A}$ *of nonempty subsets of* $\{P_1, P_2, \ldots, P_n\}$*, i.e.,* $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \setminus \{\emptyset\}$.

3.6.1 Access Tree:

Consider a tree, denoted as $\mathcal{T}$, representing an access structure. Within this structure, every non-leaf node within the tree signifies a threshold gate. Each gate is defined by its children nodes and a specific threshold value. In ABE, authorized users must satisfy the access policy formulated using a set of attributes to gain access to sensitive data. An access structure shown in Figure 3.5 gives access to those users who satisfy the access policy $(((X \wedge Y) \vee Z) \wedge V)$.

The next chapter (Chapter 4) presents a comprehensive analysis of the impact of varying workloads on performance and scalability of several existing blockchain platforms, including Hyperledger Fabric, Ethereum, and Hyperledger Besu. The chapter examines the impact of various workload parameters—namely transaction volume, transaction type, network size (number of peers), and ordering service—on system throughput, latency, and total resource utilization.

Chapter 4

Implications of Workload Dynamics on Blockchain Scalability and Performance

The purpose of this chapter * *to extensively measure and evaluate the performance of various existing blockchain platforms such as Ethereum (private Deployment), Hyperledger Besu Ethereum Client, and HLF. It is crucial to evaluate diverse performance metrics among the numerous available blockchain platforms to make a well-informed decision regarding selecting an appropriate platform for utilization.*

The results indicate that HLF outperforms both Ethereum and Hyperledger Besu. The proposed methodology offers a detailed performance evaluation of HLF, considering multiple metrics such as resource consumption, throughput, success rate, and latency. Various parameters are examined to comprehensively evaluate the system's performance, including the ordering service, programming languages used for writing chaincode, the number of transactions, transaction per second, and the number of organizations involved. In addition to the analysis, the chapter also specifies potential areas for future research and development within the blockchain landscape.

4.1 Introduction

The launch of the Bitcoin network in 2009 marked a pivotal moment in the evolution of digital currencies, popularizing blockchain, which has since been adapted and employed in numerous other cryptocurrencies [3]. Through a decentralized and distributed system,

*The research work covered in this chapter has been published in: Anita Thakur, Virender Ranga and Ritu Agarwal, "Workload dynamics implications in permissioned blockchain scalability and performance," *Cluster Computing*, Springer, vol. 27, issue. 8, pp.11569-11593, 2024, DOI: https://doi.org/10.1007/s10586-024-04550-z. **(SCIE, IF=3.6)**
&
Anita Thakur, Virender Ranga, and Ritu Agarwal, "Performance benchmarking and analysis of blockchain platforms," *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, Delhi, pp. 1-7.

blockchain enables the seamless transfer of digital assets. By utilizing cryptographic digital signatures, users can securely transfer ownership of their data while the transactions remain publicly visible on the Bitcoin blockchain. This transparency allows anyone on the network to independently verify the authenticity of transactions, ensuring trust without relying on central authorities.

One of the key advantages of the blockchain's decentralized structure is its resilience to single points of failure. With copies of the blockchain ledger stored on every node in the network, the system remains robust even if individual nodes are compromised or fail. However, scalability is one of the most pressing challenges facing blockchain networks, particularly Bitcoin. The Bitcoin network is constrained by a fixed block size limit of 1 MB, resulting in a throughput of only seven TPS. This limitation makes Bitcoin unsuitable for high-frequency trading and large-scale applications. While increasing the block size might alleviate the issue, it comes with the risk of centralization, as it would slow network propagation and heighten the storage requirements, further limiting the number of users willing to support the blockchain. Thus, addressing the delicate trade-off between block size and security is essential for the future scalability of blockchain systems.

Blockchain provides many ways to improve the data protection and verifiability of contemporary information systems. Regarding customization and potential use cases, the HLF offers a variety of functions. It permits a developer to create a unique business network on blockchain that comprises various heterogeneous clients, each having a certain purpose. The motivation behind conducting this research is to identify the multiple areas where blockchain platforms can be made more efficient, depending on the performance measure. With performance analysis, it can be easy to identify bottlenecks and factors affecting the performance of the blockchain platform.

4.2 Problem Statement

Blockchain offers a compelling alternative to centralized database systems due to its focus on security and transparency; however, its practical application is often constrained by performance issues. Inefficiencies in transaction processing and the lack of standardized protocols present significant obstacles to the broad adoption of blockchain. These limitations manifest as challenges in scalability, heightened transaction delays, and throughput bottlenecks. Blockchain faces challenges in managing the substantial transaction volume in enterprise settings. Bitcoin processes approximately seven TPS, attributed to its limited block size

and the PoW consensus mechanism. It does not meet the requirements for widespread implementation as a digital payment system. Although Ethereum operates faster than Bitcoin, it still faces congestion during elevated activity, increasing latency. Most existing research works have mainly outlined the architectural and conceptual frameworks for implementing blockchain in various domains. Successful blockchain deployment necessitates carefully considering factors, including transaction volume, rates, and participant numbers.

4.3 Research Contributions

The contribution of the proposed methodology is discussed as follows:

[1] To provide an in-depth evaluation of the performance of three leading blockchain platforms: HLF, Ethereum (privately deployed), and Hyperledger Besu. By conducting a series of performance tests, these platforms are compared based on several key metrics, including throughput, latency, scalability, and resource consumption. The findings show that HLF outperforms Ethereum and Hyperledger Besu regarding throughput and latency, positioning it as the superior choice for enterprise-level applications requiring high efficiency. It also focuses on the detailed performance analysis of HLF, offering insights into its optimization potential across various real-world use cases.

[2] The work examined the impact of increased workload, TPS, ordering services, and organizational factors on the scalability and performance of the blockchain platform.

[3] The work investigates the various errors that lead to transaction failure.

4.4 Methodology

HLF offers a broad range of configurable features and potential applications for developers to establish a bespoke blockchain business network comprising heterogeneous clients with distinct roles assigned to them. Additionally, it facilitates the creation, deployment, and execution of chaincode on the platform. The methodology adopted for the experiment is shown below in Figure 4.1. To assess and quantify the effectiveness of the HLF platform, the Hyperledger Caliper is employed (Figure 4.2), which can produce a performance report tailored to specific use cases.

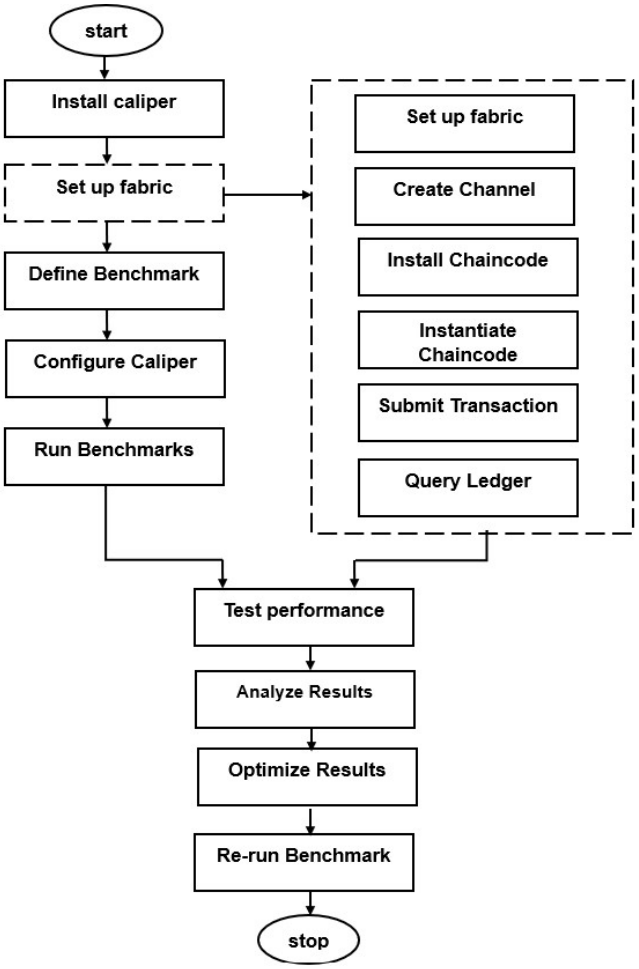


Figure 4.1: Methodology employed in the experiment

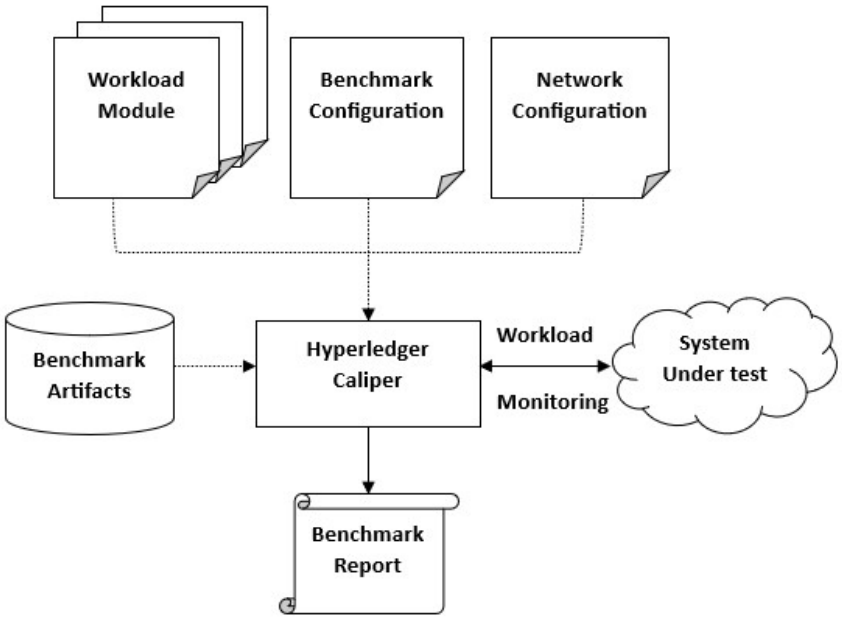


Figure 4.2: Implementation of Caliper for executing benchmark

4.4.1 HLF

Hyperledger is a collaborative open-source project from the Linux Foundation that intends to promote blockchain development across various industries [120] . It involves the participation of leaders from different sectors worldwide, and its goal is to advance the application of blockchain collaboratively and innovatively. It permits organizations to customize the technology to meet their needs and achieve maximum usability with minimal effort. HLF is a blockchain project under the Hyperledger umbrella that utilizes chaincodes and ledgers to enable participant transaction management. However, it differs from other blockchain systems because it is permissioned and private. On the contrary, to open permissionless systems that allow unknown entities to join the network, HLF requires participants to conscript through an entrusted MSP to ensure the network's security. It means that it does not rely on protocols like PoW to validate transactions and maintain network integrity. To ensure transaction privacy within the HLF network, a channel system isolates transactions and restricts access only to authorized nodes within that specific channel. The various roles concerned with the flow of transactions within a channel are described below.

4.4.1.1 Peer

Peers are a crucial component of the HLF network as they serve as hosts for ledger and chaincode instances. In blockchain, specifically on the HLF platform, a chaincode refers to a smart contract. These chaincodes are written in languages like Go, Java, or Node.js and manage the data stored on the blockchain. Essentially, they automate and enforce agreements between parties involved in a blockchain transaction. Essentially, chaincode acts as the business logic layer of a blockchain application, specifying how assets can be created, transferred, and updated on the ledger. Each peer can install multiple chaincodes and ledgers. While each peer doesn't need to install the chaincode, peers can communicate through the gossip protocol to update and modify the latest state of the ledger. It allows for efficient and decentralized communication among peers within the network. The organization peers within the HLF network can be categorized into several types: leader, anchor, committing, and endorsing peers. Each peer is assigned specific behaviors and operations within the network. A leader peer node is crucial in the network because it receives newly generated blocks from the orderer and distributes them to other peers through the gossip protocol. On the other hand, anchor peers are known to different organizations within the network, allowing them to communicate using the gossip protocol. Endorsing peers are

responsible for endorsing transactions sent to them by the ordering services. These peers play a vital role in validating the transactions before they are committed to the ledger. In contrast, committing peers do not execute transactions but receive the transactions from the orderer and commit them to their local copy of the ledger. It ensures that the network maintains a consistent and accurate record of all transactions.

4.4.1.2 Orderer

The HLF network relies on the orderer to pack the transactions into blocks and send them to anchor peers across the network. The flow of transactions within the network consists of three distinct stages: The proposal stage, the packaging stage, and the validation stage. Although the orderer is primarily responsible for the packaging step, it also contributes to the validation step by sharing new blocks with the network. By doing so, the orderer ensures that the network maintains a precise and current ledger of all transactions. In the HLF network, the orderers package transactions into blocks and send them across the network, and play a vital role in enforcing basic access control for channels. They control who has the authority to read and write data to the channels and configure them. It is important to note that the ability to modify configurable elements within a channel is subject to policies established by the relevant administrators during the creation of the consortium or the channel. By enforcing access control and adhering to established policies, orderers ensure that the network operates securely and efficiently. The Fabric 2.0 version recommends using Raft ordering services, replacing the previously deprecated Kafka ordering services.

4.4.1.3 Organization

An organization in HLF refers to a group of peers that work together towards a common goal in conducting business. These peers can comprise one or more members collaborating to maintain and operate the network. Each organization has its own MSP, which oversees its members' identity and access control. By dividing the network into separate organizations, HLF allows participants to transact and collaborate securely while retaining control over their data and operations.

4.4.1.4 Failed Transactions

Blockchain transactions may get rejected or fail to be processed due to various factors, including errors in syntax, errors in achieving consensus among nodes, and errors arising from the version of the blockchain software being employed. The consensus errors in blockchain

occur when there is a dispute between the nodes in the network about the validity of a transaction. It can transpire for various reasons, such as a network partition or a malicious node trying to manipulate the consensus [120]. In the case of HLF, validation logic errors can occur when the transaction's validation system endorsement policy is not satisfied and the peer nodes cannot reach a consensus on the transaction's validity. The chaincode defines this endorsement policy and specifies the number of required endorsements, the identities of the endorsing peers, and the criteria for successful endorsement. The transaction will not be endorsed and will be rejected if the policy is not realized. For instance, in HLF, the endorsement policy specifies the required number of signatures from endorsing peers who must agree that the transaction is valid. If the endorsement policy is not met, the transaction is deemed invalid and is rejected by the system. The policy ensures that transactions completed on the blockchain are valid and fulfill the network's predetermined requirements. Our experiment observed transaction failures related to endorsement policy, conflicts, and timeout. There are various reasons outlined for transactions to fail in the HLF blockchain, some of which are described below:

[**1**] ***Insufficient Endorsement:*** Before a transaction can be added to the ledger, it needs endorsement from a certain number of peers, and not meeting the requirement of a specified number of endorsements leads to transaction failure.

[**2**] ***Inadequate Resources:*** Due to the lack of resources available, such as processing power and network bandwidth, transactions may fail in HLF since it depends on the network of peers to validate transactions.

[**3**] ***Versioning Conflict:*** Multiple versions of chaincode deployment on the network are allowed by the HLF blockchain. However, if the transaction submitted to the smart contract is not compatible with the network's version, then the transaction leads to failure.

[**4**] ***Transaction Validation Failure:*** If the validation mechanism in the HLF smart contract is poorly described, a validation fault results in transaction failure and unsuccessful transaction validation.

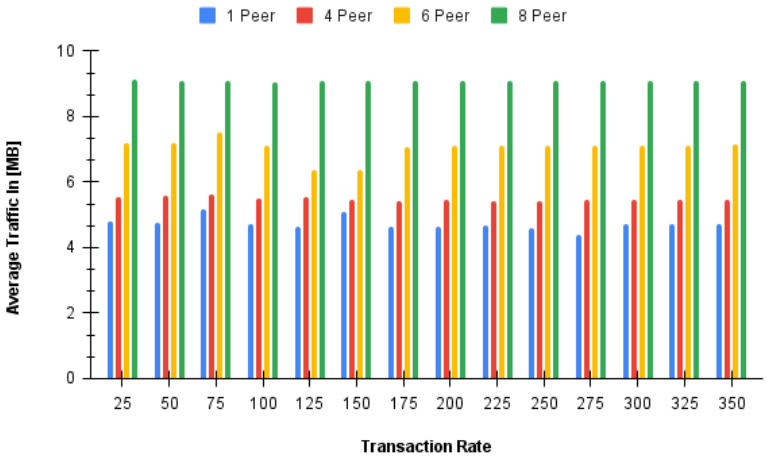
[**5**] ***Network Congestion:*** Hyperledger blockchain network communicates in a peer-to-peer manner to reach consensus; if the network is congested or encounters latency issues, the transaction in the network fails, too.

4.4.2 Performance and Scalability

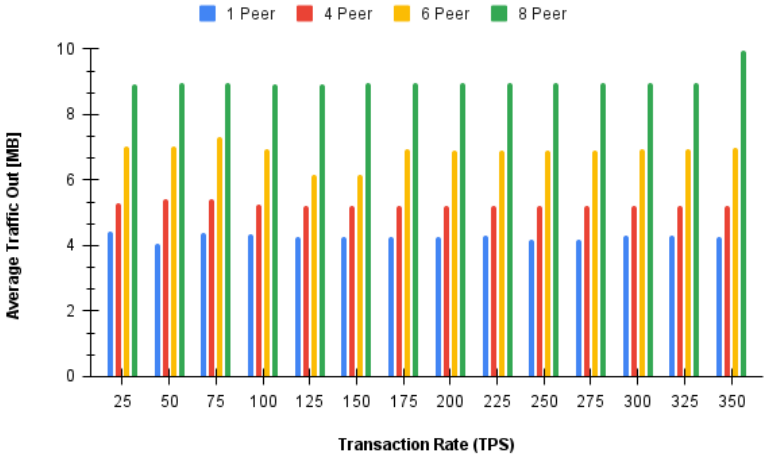
Analyzing the performance of blockchain platforms is pressing in guaranteeing the users that it can help deliver the required quality of service and meet the demands of its intended use cases. The user can distinguish limitations and bottlenecks in the scalability and performance of the system through performance analysis. The performance analysis can serve in optimizing the resource utilization of the system, such as network bandwidth, CPU, and storage, to enhance the overall efficiency [121]. Furthermore, performance analysis can assist in determining the possible security concerns and ensuring the system can manage high transaction volume without jeopardizing its integrity and security. The evaluation encompasses key metrics such as success rate, throughput, latency, and resource consumption (in detail section 2.4). The scalability of the HLF can be evaluated with varying workloads, number of transactions, organization, and ordering services. Scalability aims to increase the network's capacity to process significant transactions without compromising its security and decentralized structure.

During the test execution, transactions utilize the resources such as CPU, cache memory, disk, and RAM. Extreme consumption of these resources, e.g., disks, can significantly impact the systems' performance [11].

In network I/O, traffic in and out refers to the amount of data received by a peer or orderer node (traffic in) and the amount of data sent out from the node to other network participants (traffic out). The figures depicting the data on network input/output (I/O) traffic, measured in megabytes (MB), are denoted in Figure 4.3. The evaluation is performed with varying numbers of peers and transaction rates while keeping the transaction volume constant at 1,000 transactions. The resultant graph reveals a no trend: a demonstrable rise in average inbound traffic as the number of peers within the network increases. High Traffic In and Traffic Out indicate significant data transfer between the client and the blockchain network. It can help assess the network's capacity to handle data volume. Comparing Traffic In and Traffic Out for different workloads can reveal which scenarios generate more data transfer. It aids in understanding the impact of various use cases on the network. Several use cases are outlined by varying the key parameters, such as transaction rate and number of transactions, to perform the test on the Hyperledger Caliper. These use cases enable assessment of the system's performance and functionality under different situations and conditions, which can help identify potential bottlenecks and improve system efficiency.



(a)



(b)

Figure 4.3: Graphical representation of resource monitor of Traffic In [MB] and Traffic Out [MB] with multiple peers

4.5 Experimental Results and Analysis

In this section, the experimental results and their corresponding analysis are presented. The study is based on an evaluation of three different blockchain platforms: HLF (versions v2.2 and v2.4), Ethereum (private deployment), and the Hyperledger Besu Ethereum client (latest version), as described in Table 4.1. In private deployments, HLF frequently surpasses Ethereum (Private Deployment) and Besu (Ethereum Client) when it comes to throughput and latency, as illustrated in Figure 4.4. Here, Figure 4.4(a) shows that the HLF v2.2 and v2.4 handle all the transactions relatively well and the Figure 4.4(b) represents that the throughput for the "query" type transaction is almost equal in all the blockchain, whereas throughput for "open" type transactions and the "transfer" type transactions are quite un-

Table 4.1: Infrastructure setup and system specification

Component	**Description**
CPU	AMD Ryzen 5 5500U with Radeon Graphics
RAM	16 GB
SSD	512 GB
Operating System	Ubuntu v22.04 LTS
Blockchain Infrastructure	HLF v2.2.9, v2.4.7, Ethereum, Hyperledger Besu
Docker Engine	v20.10.21
Docker Compose	v1.28.5
Benchmarking tool	Hyperledger Caliper v0.4.2, Latest
Programming Language	Go language, JavaScript
IDE	VSCode

satisfactory. The study also illustrated that the HLF version v2.4 has increased throughput compared to the other platforms.

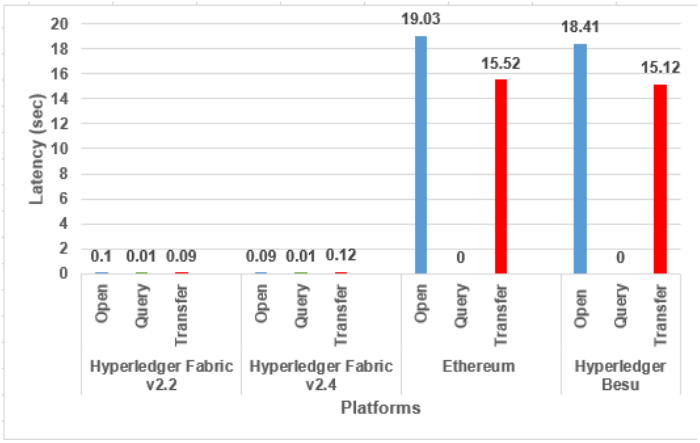
It has been observed from the study that the HLF has outperformed the other blockchain by a significant margin. In HLF, the batch size and the batch time are key factors in achieving high throughput Figure 4.5(b). The Batch Size specifies the number of transactions the orderer collects before cutting a block. In the resultant study, at the batch size with 100 message count, the latency and throughput show linear/slower growth as seen in Figure 4.5.

This research is more focused on analyzing the performance characteristics of HLF under varying workloads. This section discusses the findings of this research work and delves into the reasons behind the observed behavior. The subsequent section demonstrates the different configurable parameters cataloged in Table 4.2 along with the experimental setup specified in Table 4.3 to conduct the evaluation. The impact of varying the transaction rate (rateControl), the number of transactions, the programming language adopted to write the smart contract, and the number of organizations on the blockchain's performance is measured. A simple money transfer chaincode that incorporates the functions open() and query() is employed in the experiment. Using the open function, a new account is generated and initialized with the specified amount of money.

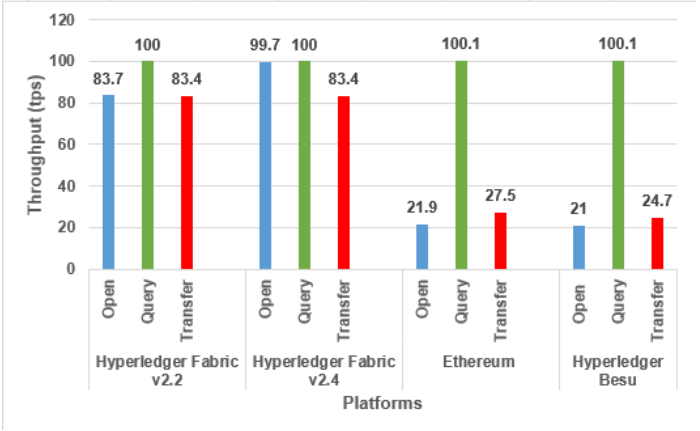
The query function retrieves the amount of money in a given account. It has been observed that the complexity of the chaincode used to execute transactions can also impact performance. More complex chaincode can require more processing power, leading to slower

Table 4.2: Configurable parameter for HLF performance analysis

Examination	Parameters	Transaction Rate (TPS)	Number of transactions (TXNs)
Impact of transaction rate TPS	Varying TPS and keeping TXNs constant	25,50,75,100,125,150,175,200,225,250,275,300,325,350	1000
Impact of TXNs (number of transactions)	Varying TXNs and keeping TPS constant	200	10, 100, 1000, 10000, 100000
Impact of ordering service	Varying TPS and keeping TXNs constant Varying TXNs and keeping TPS constant	25,50,75,100,125,150,175,200,225,250,275,300,325,350 200	1000 10, 100, 1000, 10000, 100000
Impact of programming language	Chaincode written in Go language also keeping TPS constant while varying TXNs Chaincode written in Go language also keeping TXNs constant while varying TPS	200 50, 100, 150, 200, 250, 300,350	10, 100, 1000, 10000, 100000 1000
	Chaincode written in Node.js also keeping TPS constant while varying TXNs Chaincode written in Node.js also keeping TXNs constant while varying TPS	200 50, 100, 150, 200, 250, 300,350	10, 100, 1000, 10000, 100000 1000
Impact of multiple organizations	Increased number of organizations with varied TPS and constant TXNs Increased number of organizations with constant TPS and varied TXNs	25,50,75,100,125,150,175,200,225,250,275,300,325,350 200	1000 10, 100, 1000, 10000, 100000



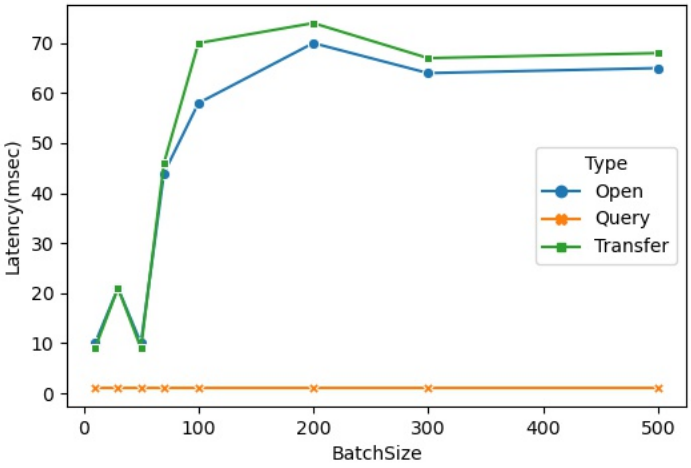
(a)



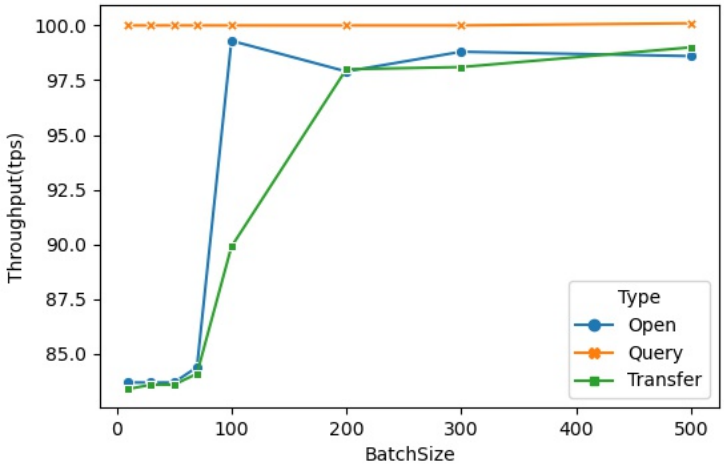
(b)

Figure 4.4: Latency (average latency) and Throughput measured for different platforms for 1000 TXNs transactions

transaction processing times. As established earlier, this research involves varying parameters such as transaction rate, ranging from 25 TPS to 350 TPS, while keeping the number of transactions fixed at 1000. Additionally, the number of transactions varies while keeping the transaction rate fixed to observe the significant changes in throughput and latency, the resource consumption of the various components, and identify potential areas for optimization in the system. By varying these parameters and observing the resulting changes in performance, this research aims to obtain a more exhaustive understanding of the system's behavior under different scenarios and conditions. The performance of the blockchain is affected not only by transaction rate and number but also by the number of peers, orderers, and organizations. In this research, the number of organizations varies from one to three, focusing on observing the impact on the performance of the blockchain. The experiment employs a multi-round test case where the transaction rate is progressively increased in each round. Furthermore, each new round has a higher transaction rate than the previous round.



(a)



(b)

Figure 4.5: Latency and throughput with varying batch size

Each experiment has been conducted at least 20 times for each defined parameter.

a. Impact of Transaction Rate on Latency and Throughput

Here, Figure 4.6 (a) illustrates the throughput and average latency for 1 organization with 1 peer and single orderer over different transaction rates, and Figure 4.6 (b) illustrates the throughput and average latency for 2 organizations with 2 peers (1 peer per organization) and single orderer over different transaction rates as well.

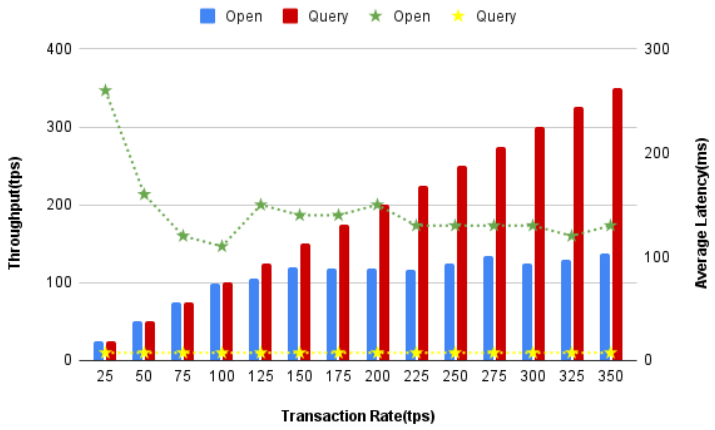
Observation: As mentioned previously, the chaincode deployed in the Fabric network is related to a simple money transfer between the accounts and has the functions open and

Table 4.3: Experimental setup and configuration for HLF performance evaluation

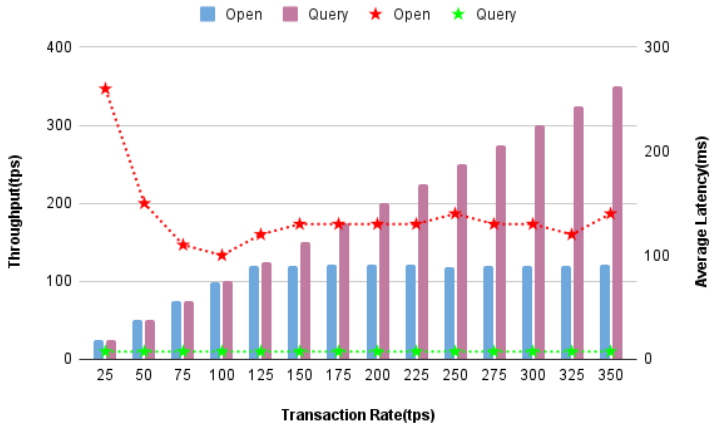
Component	**Description**	**Parameters**
Platform	HLF	v2.4.9
Number of Organizations	The number of independent organizations participating in the network	1, 2, 3
Database	State Database	CouchDB
Channel	Private communication pathway	Single private channel
Number of Ordering Service	The number of nodes participating in the ordering service (Solo, Raft)	1, 3, 5 orderer
Chaincode Implementation	The programming language used to develop the chaincode (SC)	GO, Node.js
Transaction Type	The type of transaction being executed	Open, Query
Transaction Rate	The number of transactions submitted per second to the network	25 TPS, 50 TPS, 75 TPS, 100 TPS, 125 TPS, 150 TPS, 175 TPS, 200 TPS, 225 TPS, 250 TPS, 275 TPS, 300 TPS, 325 TPS,350 TPS
Number of Transactions	The number of transactions submitted	10, 100, 1000, 10000, 100000

query. Throughput is evaluated by varying transaction rates with each round, starting from 25 transactions per second (TPS) and increasing by 25 TPS in each subsequent round until reaching 350 TPS. From Figure 4.6 (a), it can be noticed that for open( ), transaction rate ranging from 25 TPS to 150 TPS, there is a linear growth in throughput, and the throughput delivered is 25 TPS, 49.9 TPS, 74.5 TPS, 99.7 TPS, 105.7 TPS, and 119.6 TPS, respectively. However, after the transaction rate of 150 TPS, throughput decreased as the transaction rate increased. It suggests that the system may encounter bottlenecks as the load increases beyond a certain point. Based on the result, it can be inferred that the test environment used in the research can handle open() type of transactions up to a rate of 150 transactions per second (TPS). At a transaction rate of 25 TPS, the average transaction latency is higher than rates ranging from 50 TPS to 350 TPS. Specifically, the average latency at a rate of 25 TPS is around 0.26 seconds or 260 milliseconds.

According to the results presented for our test environment, the average latency of trans-



(a)



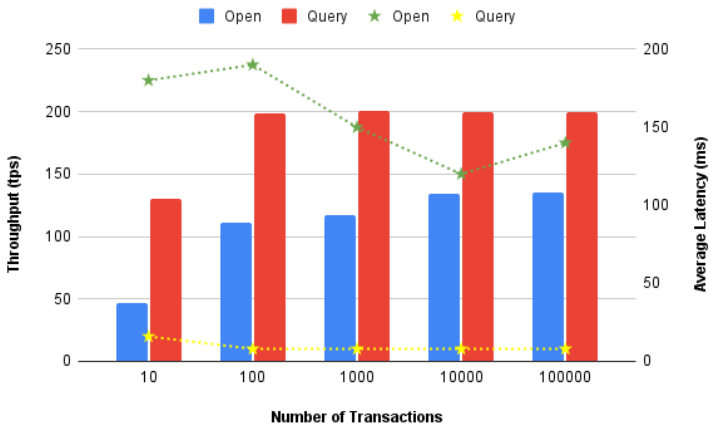
(b)

Figure 4.6: Impact of transaction rate on latency and throughput (a) 1 organization 1 peer and (b) 2 organizations 2 peers

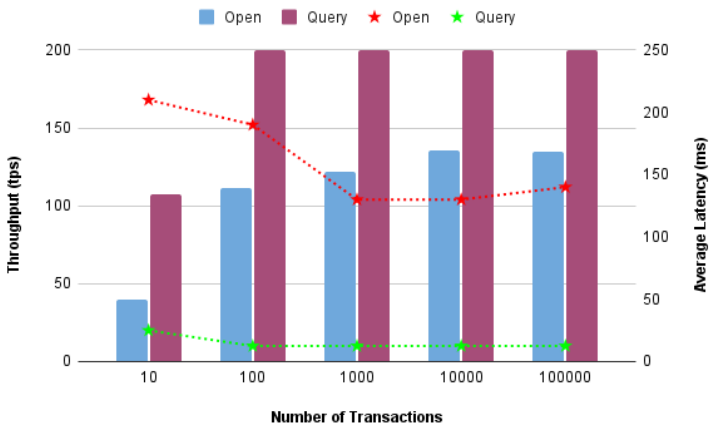
actions is at its minimum when the transaction rate is 100 TPS, with an average latency of 110 milliseconds or 0.11 seconds. After the transaction rate of 225 TPS, the average latency varies between 130 ms to 120 ms. It indicates that the system is effective when handling transactions at a 100 TPS rate and that increasing or decreasing the transaction rate may result in higher latency and can impact performance. Likewise, in Figure 4.6 (b), for open (), it can be seen that from transaction rate 25 TPS to 200 TPS, there is an increase in throughput (TPS) as compared to (a). Moreover, the average latency is relatively low in Figure 4.6 (b) until the transaction rate of 175 TPS is reached, after which there is some variation in the average latency between 130 ms and 140 ms, respectively. It has been observed that in this test environment, either one or two organizations, the query type transaction can be executed at 350 transactions per second (TPS) without experiencing any significant delays.

b. Number of Transactions' Impact on Latency and Throughput

Here, Figure 4.7 demonstrates the impact on throughput and average latency by varying the number of transactions over a fixed transaction rate. The broad array of transactions, 10,100,1000,10000,100000, are submitted to the blockchain at the fixed rate of 200 TPS. The Figure 4.7 (a) illustrates the impact on throughput and latency for 1 organization with 1 peer, and (b) depicts the impact for 2 organizations with 2 peers (1 peer per organization) by varying the workload.



(a)



(b)

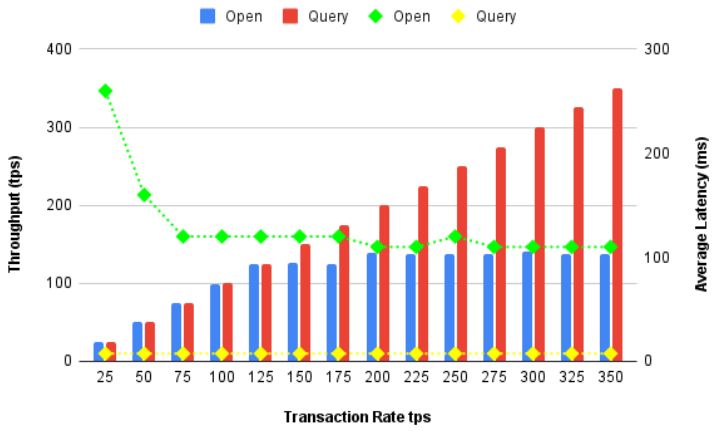
Figure 4.7: Number of transactions impacting performance from the perspective of latency and throughput (a) 1 organization 1 peer (b) 2 organization 2 peers

Observation: Here, Figure 4.7 (a) and (b) evaluate the influence of the number of transactions on average latency and throughput. For open(), 10 transactions are sent at the rate of 200 TPS (as shown in Figure 4.7 (a)), the observed throughput is 46.5 TPS, and the average latency is high, measuring 180 ms or 0.18 seconds. Similarly, in Figure 4.7 (b), throughput is observed to be 39.4 TPS, and the average latency is 210 ms or 0.21 sec. In

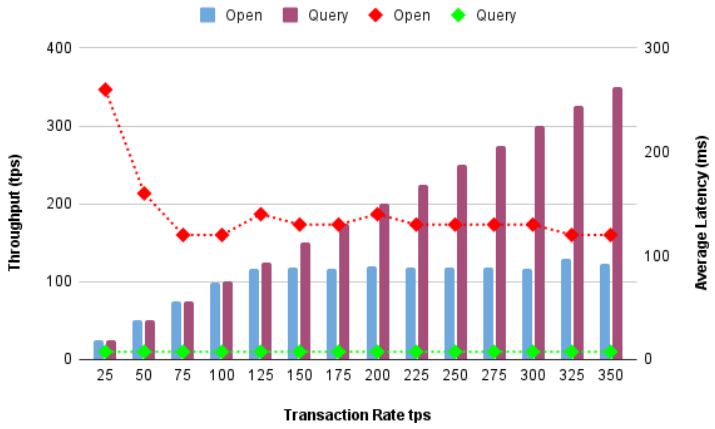
addition, the throughput for 100, 1000, 10000, and 100000 transactions for 1 organization and 2 organizations are 110.7, 117.5, 134.3, 135, and 111.2, 121.8, 135.5, 135.2 TPS, respectively. Graphical representation Figure 4.7 (a) and (b) illustrate the impact on latency and throughput when 10 transactions are sent at a rate of 200 TPS in query type transactions. The observed latency in both cases is 0.02 sec, slightly higher than the average latency observed for 100, 1000, 10000, and 100000 transactions. It indicates that the number of processed transactions impacts the blockchain system's performance. This system has been observed to support up to 100,000 transactions, depending on the function (open, query), with minimal latency.

c. Impact of Ordering Service on Latency and Throughput

The Figures presented demonstrate the impact of ordering services on average latency and throughput. The Figure 4.8 (a) illustrates the impact on throughput by employing the Raft



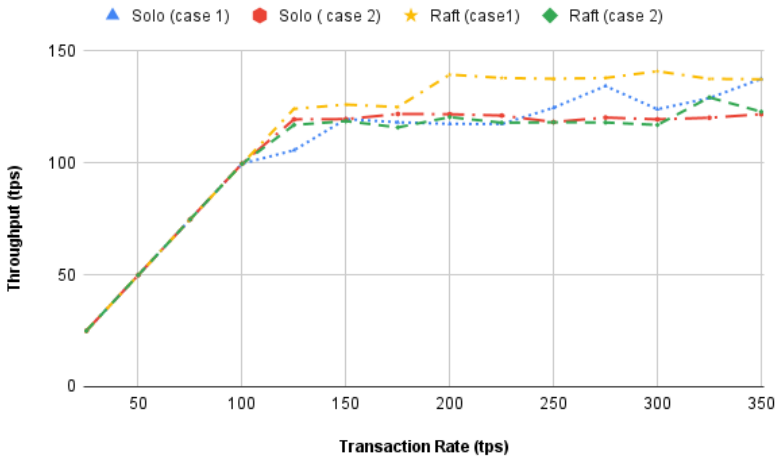
(a)



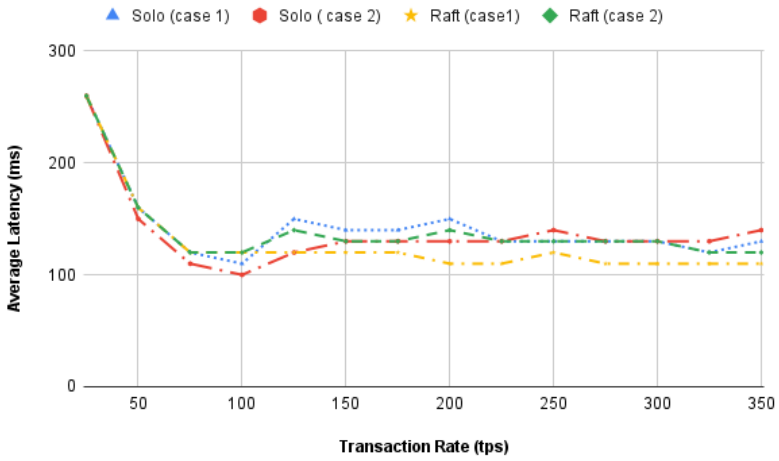
(b)

Figure 4.8: Impact of Raft ordering service on throughput and average latency

ordering service in the 1 organization 1peer, while Figure 4.8 (b) illustrates the same for the 2 organizations 2 peers scenario. Graphical representations Figure 4.9(a) and Figure 4.9(b) show the variation in throughput and latency exerting the Solo ordering service in 1 organization 1 peer (case 1), 2 organization 2peer (case 2), and Raft ordering service in 1 organization 1peer (case 1) and 2 organization 2peer (case 2).



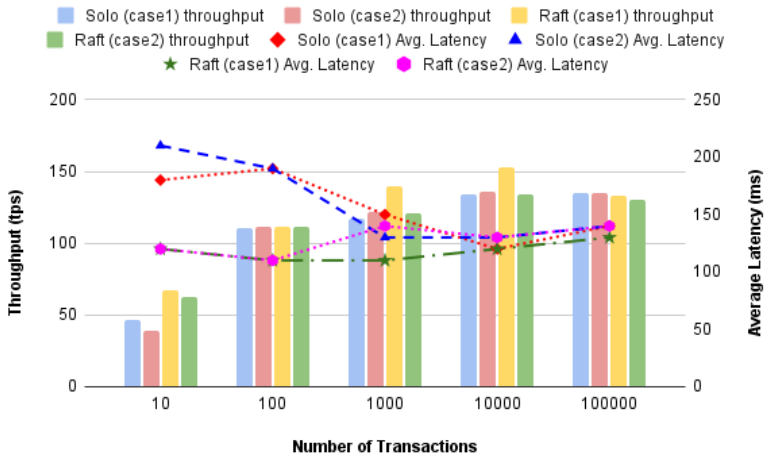
(a)



(b)

Figure 4.9: Impact of Solo and Raft ordering service on throughput and average latency

Observation: It can be observed from Figure 4.8(a) for open() that there is a linear growth in throughput with the varying range of transaction rate from 25 TPS to 150 TPS and again shows a rise in throughput can be seen at 200 TPS. On the other hand, the observed average latency in the same scenario is higher at 25 TPS and 50 TPS than the transaction rate, ranging from 175 TPS to 350 TPS. In contrast to the results shown in Figure 4.8(a), the average latency observed for transaction rates ranging from 75 TPS to 350 TPS is between



(a)

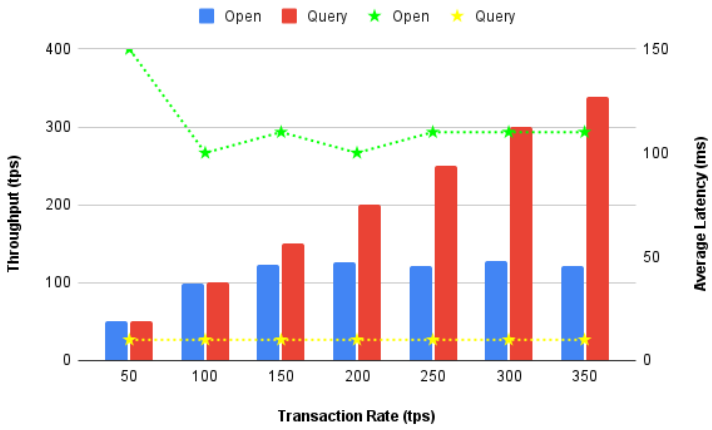
Figure 4.10: Impact of Solo and Raft ordering service on throughput and average latency with varying number of transactions

0.12 seconds to 0.11 seconds, indicating a lower latency compared to the results obtained in Figure 4.8(b). The Figure 4.9(a), Figure 4.9(b), and Figure 4.10 figures describe the impact of employing Solo and Raft ordering services on throughput and latency. It can be observed that the Raft ordering service handles up to 100,000 transactions very effectively with minimum delay. The results obtained from our test environment show that the Raft ordering service performs effectively in the context of increased throughput and less latency compared to the Solo ordering service in both cases, i.e., with varying transaction rates and numbers of transactions.

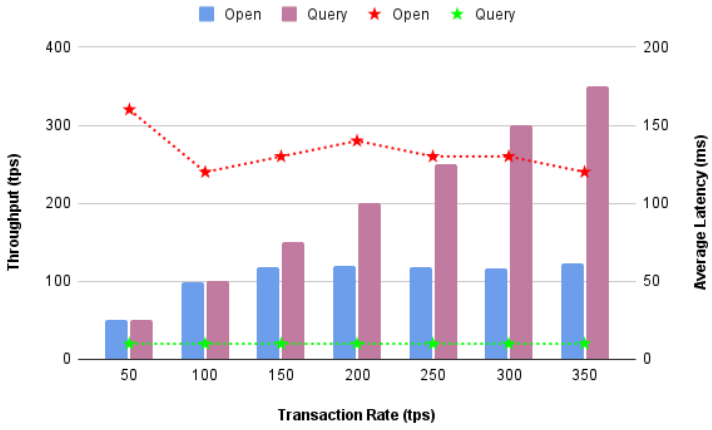
d. Impact of Programming Language on Latency and Throughput

In the HLF blockchain, a chaincode is a program or a smart contract that establishes the business logic and rules governing the interactions between participants in a blockchain network. Chaincode is deployed on a specific channel of a blockchain network and is utilized to read and write the ledger's state. Chaincode in the HLF blockchain can be written in Node.js or Golang. The chaincode employed in this research is written in Go language and Node.js and evaluated to see the impact on latency and throughput under different transaction rates (Figure 4.11) and the number of transactions (Figure 4.12)).

Observation: Here, Figure 4.11(a) depicts the performance of a smart contract written in the Go language, while (b) shows the performance of a smart contract written in Node.js. In Figure 4.11(a), (b), for open() type transactions, there is a linear growth in throughput as the transaction rate increases from 25 TPS up to 200 TPS, implying that the system



(a)



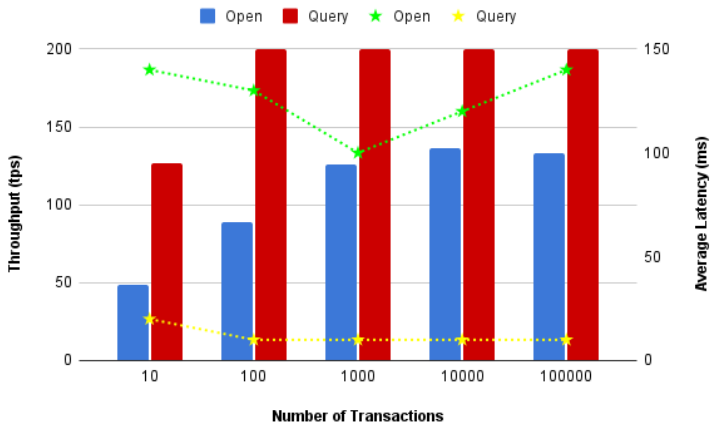
(b)

Figure 4.11: Chaincode Written in Go language(a) and Node.js (b) to evaluate latency and throughput experienced under different transaction rate

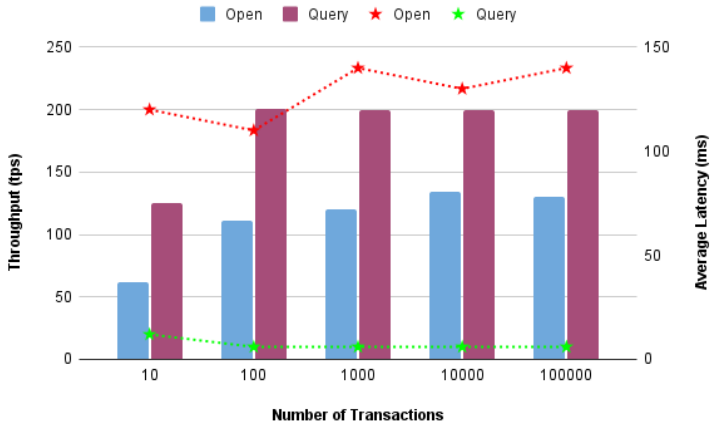
is capable of efficiently processing transactions up to a rate of 200 TPS. According to the results (Figure 4.11 and Figure 4.12), the chaincode developed in the Go language outperforms the one developed in Node.js in the context of throughput and average latency. However, the difference is not very significant. The Go language-based chaincode shows slightly lower latency and higher throughput compared to the Node.js based chaincode. Go language is considered more efficient for chaincode development in HLF due to its static typing, compiled nature, and optimized runtime performance. Node.js, on the other hand, Node.js is dynamically typed and interpreted, which may result in lower performance and scalability for complex chaincode applications.

e. Impact of multiple organizations on Latency and Throughput

In this experiment, 3 organizations with one peer in every organization are observed in terms of throughput and latency. Graphical representation Figure 4.13 (a) and (b) depict the



(a)

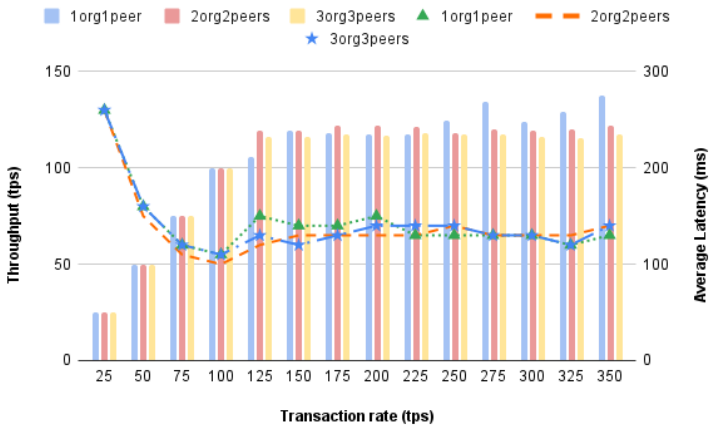


(b)

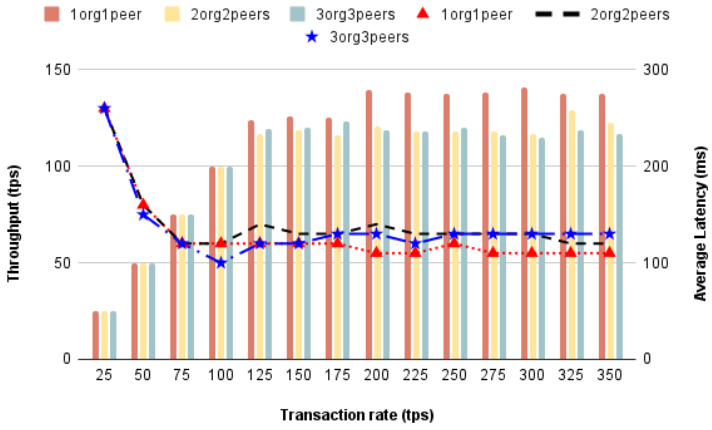
Figure 4.12: Chaincode Written in Go language (a) and Node.js (b) to evaluate latency and throughput experienced under varying number of transactions

variation in throughput and latency as the number of organizations increases. Additionally, Figure 4.14(a) and (b) present the findings concerning throughput and latency for different numbers of transactions across multiple organizations. In Figure 4.13(a) and Figure 4.14(a) Solo ordering services is employed, while Figure 4.13(b) and Figure 4.14(b) employ Raft ordering services.

Observation: Graphs illustrate that including a third organization results in linear growth in throughput for transaction rates ranging from 25 to 175 TPS and can also handle 100,000 transactions with a small delay. The results show that including a third organization has decreased throughput compared to the scenarios with a single organization and two organizations. Additionally, after reaching 175 TPS, no significant change in the throughput is observed. Graphical representation Figure 4.13 (a), (b), and Figure 4.14(b) show an increase in the average latency, indicating that the inclusion of a third organization in the



(a)

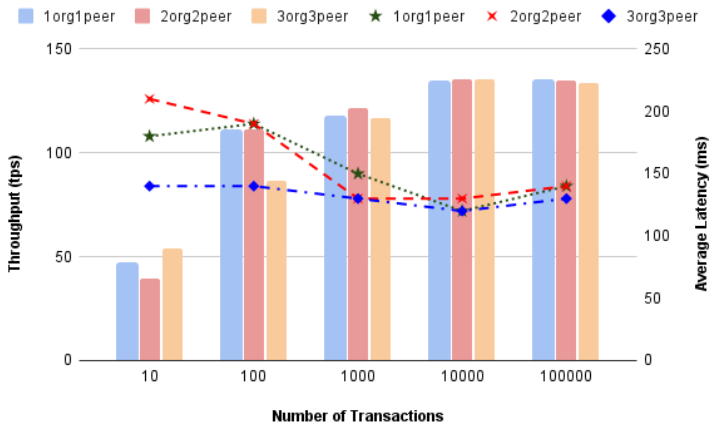


(b)

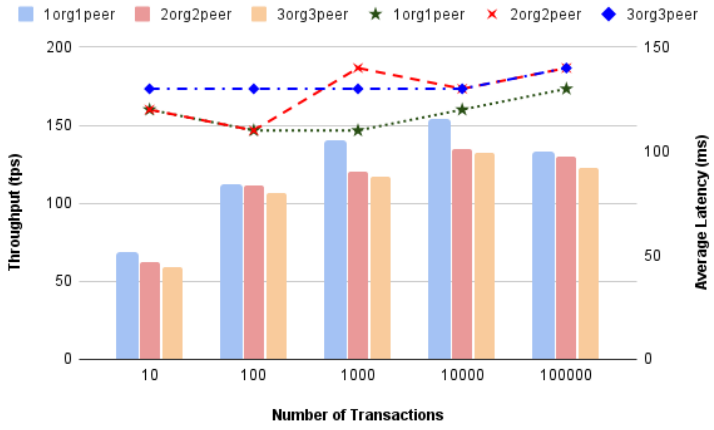
Figure 4.13: Impact of increased number of organizations on latency and throughput experienced under different transaction rates

network results in higher latency. It can be implied from the results that adding more organizations to a network can potentially result in increased complexity and communication overhead, which may negatively impact the network's performance in terms of latency and throughput.

When evaluating the performance of a blockchain network using the success rate as a measure, notable findings indicate that both the open and query operations have consistently achieved a 100% success rate across all evaluation parameters. It suggests that the permissioned blockchain under analysis has consistently executed both open() and query() without encountering any failures or errors. Based on the observations from Figure 4.15, it has been noticed that the CPU resource utilization during the execution of different workloads shows that in case 1(Using Solo ordering service with 1 organization and 1 peer), case 2 (Using Raft ordering service with 1 organization and 1 peer), case 3(Using Solo ordering service



(a)



(b)

Figure 4.14: Impact of increased number of organizations on latency and throughput experienced under varying number of transactions

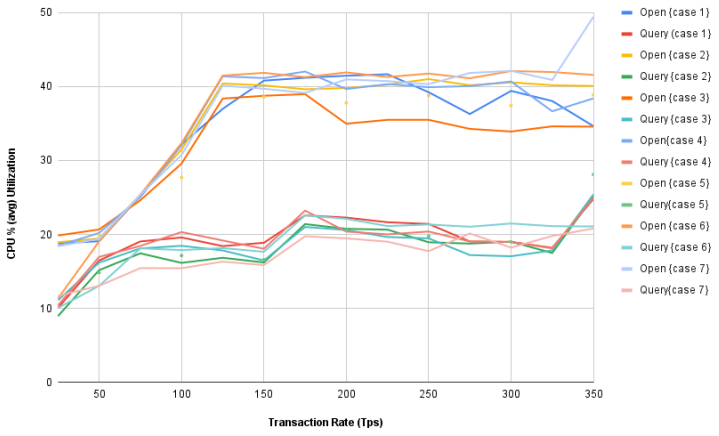


Figure 4.15: CPU% average utilization in executing open() and query() in a varied number of transaction rates for different numbers of organization and ordering services

with 2 organization with 1 peer per organization), and case 4 (using Raft ordering service with 2 organizations with 1 peer per organization), as well as case 5 (using the Go language to write chaincode), and case 6 and case 7 (using 3 organizations with 1 peer per organization), the CPU usage remains below 50%. The CPU usage for these scenarios, including configurations and setups, is relatively low and does not exceed half of the available processing power. It indicates that the executed workloads do not heavily strain the CPU resources, suggesting that the system has ample capacity to handle the workload without reaching its maximum processing capability.

HLF is a widely adopted and compelling choice for blockchain-based applications due to its modular architecture, support for confidential transactions, and scalability features. HLF has its shortcomings. HLF uses Optimistic Concurrency Control (OCC), which simulates transactions on individual peers before endorsement. It can lead to failures if multiple transactions try to modify the same data concurrently. Also, failures due to invalid signatures, mismatched world states, or policy misconfigurations can hinder successful transactions. During experimentation, instances of transaction failure were noticed, such as MVCC_READ_CONFLICT, ENDORSEMENT_POLICY_FAILURE, and the timeout expired.

The read-write conflict arises when multiple transactions try to modify the same data concurrently. Let Txn_i and Txn_j be transactions, and their respective read sets and write sets after endorsement simulation are represented as $Read(Txn_i)$: Read a set of transactions Txn_i. $Write(Txn_i)$: Write set of transaction Txn_i. Assuming Txn_i is ordered before Txn_j within the same block (i.e., Txn_i is confirmed before Txn_j), then Txn_j is considered a conflicting transaction and will be marked invalid during validation if any of the read-write conflict or write-write conflict conditions hold. HLF utilizes the Multi-Version Concurrency Control (MVCC) mechanism to manage the consistency of records. It involves tracking changes to a record's version through a structured format of key-value pairs and corresponding version information, e.g., key: Value, version.

As shown in Figure 4.16, when the first transaction reaches the validation stage, MVCC detects a mismatch. The version of the key read earlier doesn't match the current version (due to the modification of the second transaction). This mismatch triggers the MVCC_READ_CONFLICT error. In HLF, transactions require endorsement from peers before being committed. The endorsement policy defines which organizations or specific peers must endorse the transaction to be valid. There could be various reasons for the

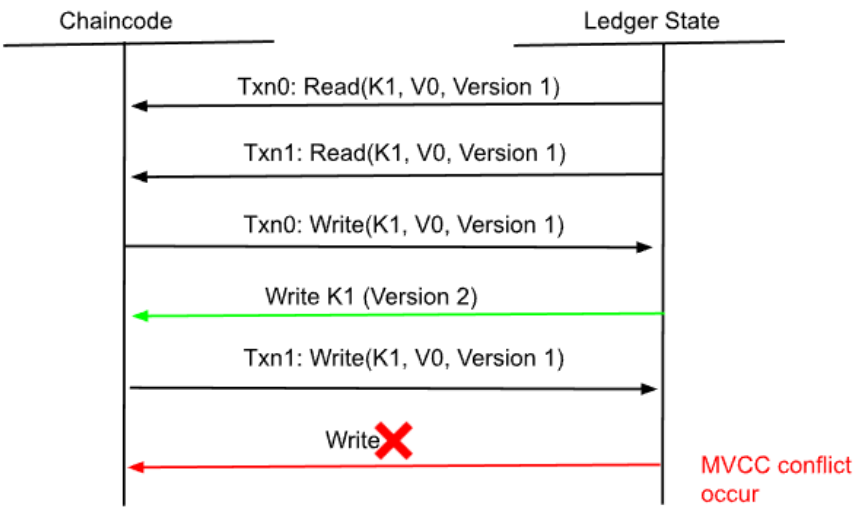


Figure 4.16: An example to illustrate read-write conflict

endorsement policy failure, such as missing endorsement, incorrect chaincode implementation, and read-write set mismatches, as concurrent writes can cause version inconsistencies leading to endorsement failure. To resolve these failures, the policy needs to be ensured to align with the transaction's participants, Caliper's features must be utilized to verify endorsement targets and chaincode logs must be examined for errors.

4.6 Discussion

Throughout the research, various factors have been analyzed that may influence the performance of the HLF blockchain. It has been observed that the number of organizations, orderers, and peers has a substantial impact on performance and scalability. A network with too many peers and organizations can result in slower transaction processing times, whereas a network with too few peers can lack fault tolerance. The transaction rate and the number of transactions influence the network's performance. As the transaction rate increases, there is a linear growth in throughput until a certain point. Our deployment of HLF encountered a critical performance bottleneck during a recent load test. As we scaled transaction volume beyond one million, many transactions began failing with a "timeout expired while executing transaction" error. This behavior suggests the network's capacity to process transactions is exceeded, leading to timeouts. Similarly, increasing the transactions per second (TPS) beyond a certain threshold resulted in the same error.

HLF offers different ordering services, such as Solo, Kafka, and Raft, each with strengths and weaknesses and features depicted in Table 4.4. In HLF (HLF), a transition from a Kafka-

based ordering service to a Raft-based consensus mechanism is implemented, commencing with version 2.0. This strategic shift aimed to streamline the ordering service and lay the groundwork for the future integration of Byzantine Fault Tolerance (BFT) capabilities. While neither Raft nor Kafka inherently possesses Byzantine fault tolerance properties, the Raft algorithm's design facilitates the inclusion of such mechanisms in future versions of HLF. It establishes the basis for potentially enhancing the consensus protocol within HLF to incorporate Byzantine fault tolerance, particularly for permissioned blockchain deployments. Our evaluation reveals that the Raft ordering service demonstrably outperformed the Solo approach in terms of performance when processing a diverse range of transaction volumes and rates.

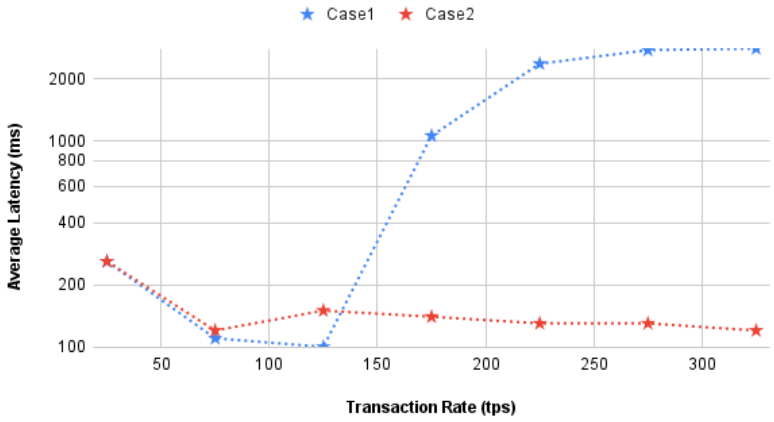


Figure 4.17: Impact of network bandwidth on latency

Further considering the impact of programming language on throughput or latency, no substantial variation exists since the programming languages Node.js and Go language have their advantages and shortcomings. However, the chaincode complexity certainly impacts the HLF blockchain performance, considering that the complex chaincode may necessitate more computing resources, resulting in slower transaction processing times. The hardware resources available to network nodes can also impact performance. Nodes with faster CPUs, larger memory, and quicker disk I/O can quickly process transactions and handle more concurrent requests. In this research, it is observed that network bandwidth plays a significant role in influencing latency. Specifically, the available network bandwidth directly impacts the time transactions are transmitted and processed within the HLF network.

In Figure 4.17, Case 1, the impact of operating within a restricted network bandwidth, which resulted in higher latency, is compared to Case 2, where the system is evaluated under conditions of high network bandwidth. HLF relies on a substantial amount of network

Table 4.4: Comparison of key features of various consensus mechanisms in ordering service

Features	**Solo**	**Kafka**	**Raft**	**PBFT**
Byzantine Fault Tolerance	No	No	No	Yes
Crash Fault Tolerance	No	Yes	Yes	Yes
Network Partition Tolerance	No	High	High	Medium
Configuration Complexity	Low	Medium	Low	High
Performance	Limited	High	High	Medium (Increased overhead)
Scalability	Medium	High	High	Medium

bandwidth to ensure efficient operation. This necessity arises from the requirement of distributing transactions to all nodes in the network and validating them.

When the network experiences congestion or has limited bandwidth availability, it can significantly impact transaction processing speed and introduce latency issues.

4.7 Summary

Blockchain is experiencing remarkable growth in interest among technology enthusiasts. Various business applications integrate blockchain to improve their offerings, including digital identification, secure information exchange, regulatory compliance, asset tokenization, anticounterfeiting, and contract management. Selecting the right platform that meets the specific requirements and delivers optimal performance across all dimensions. This chapter explores how various workload characteristics influence the average latency, resource consumption, and throughput of HLF. The results indicate that the system operates efficiently for average latency and throughput within a transaction rate range of 150 TPS to 200 TPS, successfully managing 100,000 transactions with negligible delay.

The design of HLF is fundamentally grounded in the principles of the Deterministic Consensus Algorithm. It indicates that when the ordering node transmits a new block to any peer for validation, it is guaranteed to be both definitive and precise. This design principle guarantees the uniformity and dependability of the blockchain ledger, as every node in the network agrees on the ledger's status. The evaluation noted that the Raft ordering service surpassed the Solo ordering service, showcasing superior throughput and reduced latency.

Our investigation sheds light on the impact of various factors on the scalability and per-

formance of HLF version 2.4.9. The HLF is a prominent blockchain platform that creates enterprise-level distributed ledger applications. Furthermore, an exploration into the scalability of the network revealed that with the rise in the number of organizations within the HLF network, there is a corresponding decline in throughput and an increase in latency. This chapter presents performance and scalability metrics data, including average latency, resource consumption, and throughput. In this exploration, it is concluded that several other factors also have a notable influence on the performance of HLF. These factors include CPU power, disk space, the design of chaincode, and network bandwidth.

The next chapter provides a comprehensive understanding of the proposed encryption and data access control scheme, BloCPABE. It details the foundational concepts and the concrete construction of the scheme, including its architecture, key components, and operational workflow. The chapter also defines the underlying threat model and outlines the security assumptions considered during design. Furthermore, it rigorously analyzes the security properties of BloCPABE, demonstrating its resilience against various potential attacks while ensuring fine-grained access control in blockchain-based environments.

Chapter 5

Revocable and Privacy-Preserving Data Access Scheme in Blockchain

The purpose of this chapter * *is to introduce a privacy-preserving data access approach. Technology advances are transforming healthcare into patient-centered smart systems. The mHealth uses wearable sensors, telecommunications, and IoT to create a patient-centered healthcare model that allows real-time monitoring, personalized interventions, and improved access to care, promoting proactive health management and better patient outcomes. Due to the large volume of sharing and storage of sensitive information in centralized or distributed storage, there is a growing necessity for employing encryption techniques that provide promising security and access control to sensitive data on these platforms. ABE is a cryptographic technique that offers a fine-grained access control mechanism by associating attributes with encrypted data and secret keys. This chapter introduces a revocable and secure fine-grained access scheme using blockchain and CP-ABE. The results demonstrate the relative performance of the proposed scheme, showing a significant reduction in computational costs.*

5.1 Introduction

Security attacks in the blockchain ecosystem can have profound repercussions. They can alter data, enable double spending, lead to fund theft, disrupt the network's operation, and, most significantly, erode trust in the technology. These attacks compromise the core principles of blockchain, such as immutability, transparency, and security, making it essential for

*The research work covered in this chapter has been accepted in: Anita Thakur, Virender Ranga and Ritu Agarwal, "Revocable and Privacy-Preserving CP-ABE Scheme for Secure mHealth Data Access in Blockchain," *Concurrency and Computation: Practice and Experience*, Wiley, vol. 37, p. e70064, 2025, DOI: https://doi.org/10.1002/cpe.70064 **(SCIE, IF=1.5)**,
&
Anita Thakur, Virender Ranga and Ritu Agarwal, "Attribute-based encryption scheme for secure and efficient access in blockchain", *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, Bangalore, pp. 653-658.

blockchain networks to implement robust security measures to mitigate these threats and maintain their integrity. The healthcare sector generates substantial critical data, presenting valuable research opportunities. However, a primary challenge within the mHealth ecosystem lies in safeguarding the inherent sensitivity of this health information. The potential threat of unauthorized access or attacks necessitates robust security measures to mitigate the risk of data breaches caused by internal and external actors [122].

Integrating mHealth devices with EHRs presents opportunities and challenges for patient privacy. EHR, a comprehensive digital record of an individual's medical history, contains highly sensitive information such as diagnoses, allergies, medications, and immunization dates. This data is directly linked to a patient's real identity, making it crucial to ensure appropriate access controls. However, patients may only sometimes desire to share their mHealth data history with every healthcare provider. Historical health data may be sensitive or irrelevant to the current medical condition. Therefore, a key concern within the mHealth ecosystem is the ability of patients to exert granular control over their health information [123]. To mitigate this challenge, a pioneering approach termed ABE [67] has garnered

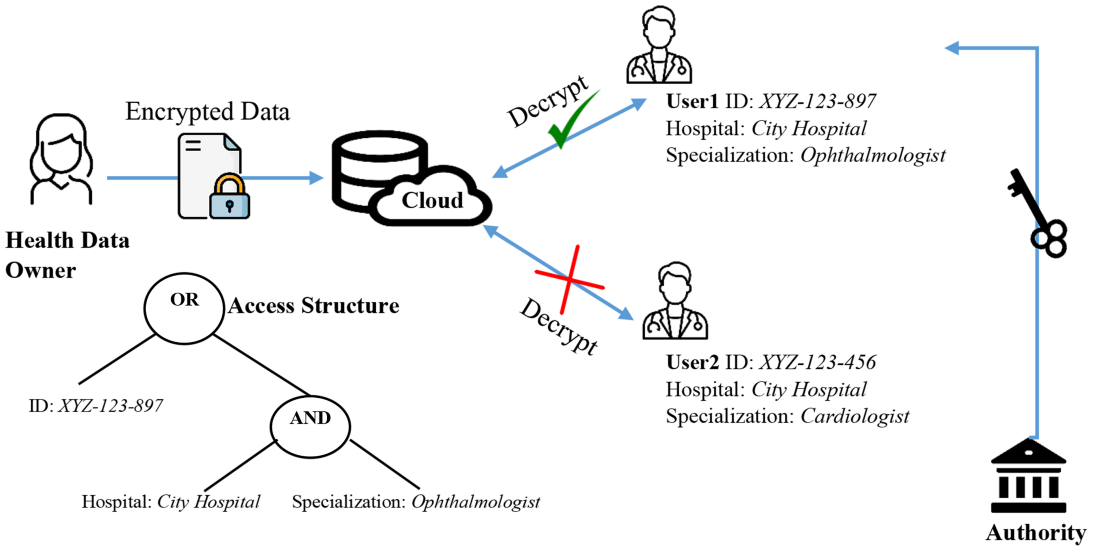


Figure 5.1: Cloud-based health data storage system employing CP-ABE scheme

significant attention. ABE furnishes a scalable framework for enforcing detailed access control on encrypted data. ABE offers a granular approach to access control for encrypted data [124]. Employing the attributes and access policies built into secret keys and ciphertexts is possible. In CP-ABE, users' secret keys are correlated with a specific set of attributes. Meanwhile, a sender generates ciphertexts accompanied by access policies dictating the required attributes for decryption a [46] as illustrated in Figure 5.1.

5.2 Problem Statement

CP-ABE provides fine-grained access control for secure cloud data sharing, but it is not easy to implement. As a central authority has access to sensitive user attribute information, it raises privacy concerns and increases the danger of misuse. In traditional public key encryption, user access can be revoked by excluding their keys, but this approach poses significant challenges in ABE. In ABE, blocklisting a user's key would necessitate revoking all keys associated with that user's attributes, which could inadvertently affect other users sharing those attributes. It would require a complete regeneration and redistribution of keys, along with re-encrypting all ciphertexts, making it impractical. In contrast, our BloCPABE framework implements an effective attribute-based revocation mechanism that seamlessly updates a user's key upon attribute revocation. It ensures both forward and backward secrecy, allowing the system to maintain secure access control without requiring extensive re-encryption or key redistribution.

5.3 Research Contribution

Over recent years, the widespread adoption of modern medical sharing systems has facilitated convenient access to healthcare services. Due to its inherent benefits, ABE has emerged as an effective strategy for securing these systems. However, the practical deployment of ABE-based medical sharing raises significant concerns regarding user privacy, data confidentiality, and overall system security. Data users seek anonymity regarding their access patterns, even from issuing authorities.

This work presents a data access control scheme that is both expressive and secure. A new BloCPABE scheme is proposed to achieve revocation and privacy-preserving capabilities. The proposed BloCPABE scheme introduces anonymous key generation, enhancing user anonymity and mitigating collusion attacks. A fine-grained data access control methodology is designed that leverages CP-ABE in conjunction with blockchain. The key contributions of this research work are demonstrated below:

[1] A secure CP-ABE scheme, BloCPABE, is proposed to address security challenges in mHealth systems. The scheme enables efficient attribute revocation by updating the associated secret key and ciphertext, requiring minimal computational overhead.

[2] The proposed scheme establishes user privacy, backward and forward security, and is

resistant to collusion attacks.

[3] Blockchain is leveraged to store outsourced data, ensuring immutability and tamper-proof integrity.

[4] The scheme incorporates the LSSS to formulate access structures, ensuring the confidentiality of the message and preventing private information leakage.

[5] Security measures and the proposed methodologies are rigorously evaluated and validated using the Scyther tool.

5.4 Basic Construction of BloCPABE

This section presents a concise overview of our proposed framework. The system model depicted in Figure 5.2 describes a secure and decentralized data-sharing framework that integrates ABE with IPFS and blockchain to enforce fine-grained access control over encrypted data. Also, Table 5.1 provides a list of the key notations used in this chapter, along with their respective descriptions. At the core, the CA initializes the system by generating system parameters and registering participants, such as the AA and Data Users. The AA manages attributes and keys, including generating public keys and issuing secret keys to users based on their assigned attributes. A DO defines an access policy that determines who can access the data. The DO encrypts the data using ABE under the policy and uploads the ciphertext to the IPFS for decentralized storage. To ensure data integrity and tamper-proof access, the IPFS hash of the ciphertext is stored on the blockchain. The proposed system model outlines the operational framework of the scheme, providing a detailed evaluation of its components as follows.

5.4.1 System Model

Here, all the participating entities and their contribution in this system model are described as:

[1] **Certificate Authority :** CA operates as a globally recognized root of trust. Its primary function is to establish and manage identities within a system. It assigns unique identifiers to the users and the authority. Additionally, the CA assumes responsibility for system initialization, including user and AA registration and disseminating global public parameters and a master secret key. While possessing a privileged role, the CA is functionally constrained, acting as an "honest-but-curious" entity with no direct

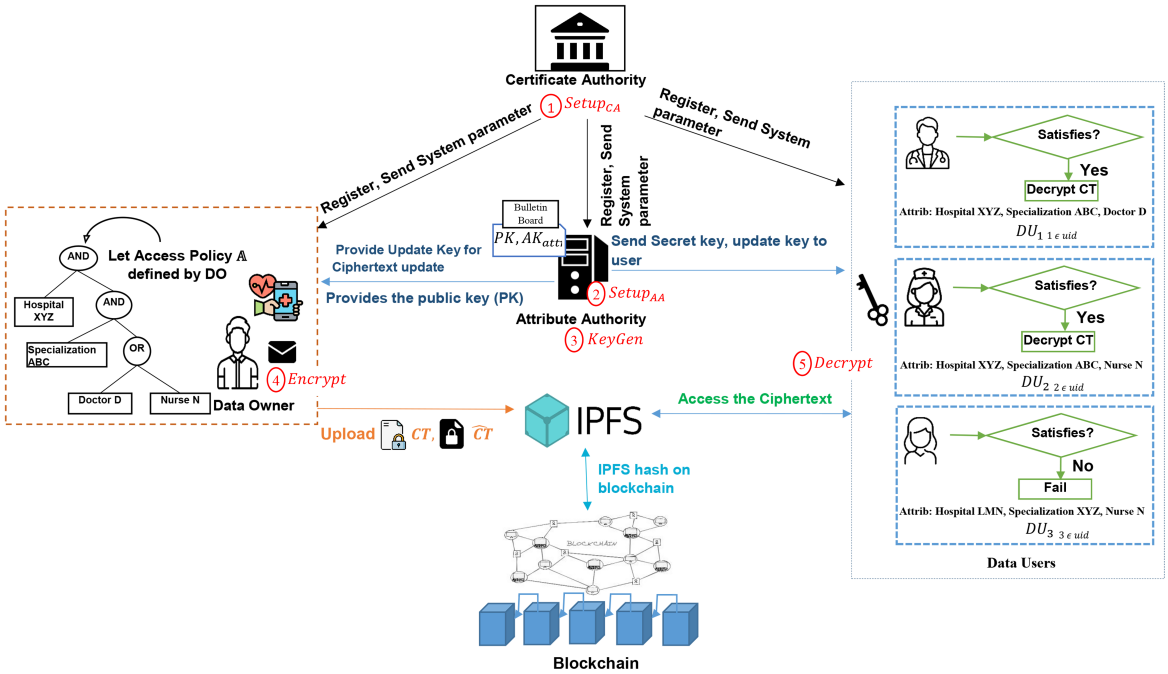


Figure 5.2: Overview of system model

involvement in attribute management or generating attribute-associated secret keys.

[2] **Attribute Authority:** AA exercises exclusive control over the management of attributes. It encompasses the issuance, updating, and revocation of attributes. AA is responsible for constructing the users' secret key.

[3] **IPFS**: IPFS is a decentralized file storage network that uses content addressing to identify and retrieve files. Instead of traditional location-based addressing, where files are accessed by their storage location, IPFS retrieves files based on a unique hash derived from their content. The encryption process generates ciphertext and associated metadata, including a policy string that governs access control.

The encrypted data is then serialized into .JSON format before being uploaded to IPFS. IPFS, a decentralized peer-to-peer file system, assigns a unique content identifier (CID) to the serialized data. This CID is a reference for retrieving the data from the IPFS network, ensuring its availability across a distributed network of nodes.

[4] **Blockchain:** Blockchain is a decentralized and immutable ledger designed for recording transactions and tracking assets. The IPFS CID is recorded on the blockchain through a smart contract. This contract stores metadata related to the encrypted data, such as its IPFS hash, ensuring its integrity and existence are verifiable. Recording this metadata on the blockchain guarantees that the reference to the data remains accurate and tamper-proof.

Table 5.1: Notations and their description

Notation	**Descriptions**
CA	Certificate Authority or Central Authority
AA	Attribute Authority (Hospital)
DO	Data Owner (Patient or Health data owner)
DU	Data User (Physician, Researcher, etc.)
$\mathfrak{A}$	Adversary
$\mathfrak{C}$	Challenger
λ	Security Parameter
G_1, G_T	Multiplicative cyclic group of prime order p
g	Generator of G_1
e	Mapping function $e(G_1, G_1) \rightarrow G_T$
$\mathcal{H}$	Collusion resistant hash function
PP	Public Parameter
GPP	Global Public Parameter
MK	Master Key generated by the CA
uid_k	Unique identifier for user k
$Certificate(uid_k)$	User's certificate
$UPK_{uid_k}, UPK'_{uid_k}$	User Public Key
$USK_{uid_k}, USK'_{uid_k}$	User Secret Key
MSK	Master Secret Key generated by AA
PK	Public Key generated by AA
$\mathcal{U}$	Universe of attributes controlled by AA
VK_x	Version Key of the attribute
AK_x	Public Attribute Key of the attribute
S	Attribute sets of user
SK	Decryption Key for DU generated by AA
$\mathbb{A}$	Access Structure
CT	Ciphertext
$\widehat{x}$	Revoked attribute from user
$AUK_{\widehat{x}}$	Attribute Update Key
$KUK_{\widehat{x},k}$	user Key Update Key
$CUK_{\widehat{x}}$	Ciphertext Key Update

[5] Data Owner: In the system, DO is responsible for storing data on the decentralized storage and sharing it with data consumers or users. The owner defines an access policy, encrypts the data according to this policy, and then sends the encrypted data to the centralized storage for secure and controlled access.

[6] **The Data User :** DU is the data consumer who wants to get the data. CA assigns the *uid* to the user. If the attributes of the DU align with the stipulated access policy, the message can be successfully retrieved.

5.4.2 Algorithm Description

Our BloCPABE framework is structured into distinct phases, each incorporating specific methods as briefly illustrated below:

Phase 1: Setup

$Setup(1^{\lambda})$: 1^{λ} here is a security parameter which *Setup* algorithm takes as input and generates global public parameter GPP and master key MK as output. Meanwhile, the CA registers authority and users, and AA also performs the setup to generate public (PK) and secret keys (MSK).

Phase 2: Secret Key Generation

$KeyGen(GPP, MSK, S_k)$: The algorithm takes the GPP, MSK, and attribute assigned to the user and generates the corresponding secret key (SK_{uid_k}).

Phase 3: Encryption of message

$Encrypt(GPP, PK, \mathbb{A}, msg)$: The algorithm requires the GPP, public key, an LSSS access structure $\mathbb{A}$ and message msg as input and generates ciphertext CT.

Phase 4: Decryption Process

$Decrypt(CT, USK'_{uid_k}, SK_{uid_k})$: The algorithm takes CT, the global secret key of user, and SK_{uid_k} as input and returns the message.

Phase 5: Attribute Revocation

$Revoke(\widehat{x}, uid_k)$: This algorithm includes the update keygen algorithm ($UKeyGen$), secret key update ($Update_{SK}$), and ciphertext update algorithm ($Update_{CT}$). The attribute $\widehat{x}$ of user uid_k is revoked, and a new secret key and ciphertext are generated for the non-revoked users.

5.4.3 Threat Model and Security Goals

CA and AA are assumed to be fully trusted. In the case of a malicious DU, unauthorized DUs might collude to gain access to the encrypted data.

[1] **Fine-grained access control:** The data owner creates a detailed access structure based on specific attributes to control access for retrieving health data. This framework guarantees that only authorized DUs have access to the encrypted information. Unauthorized DUs, regardless of how they combine their attributes and keys, cannot extract confidential details from the encrypted data.

[2] **Confidentiality:** In this system, data access is restricted to devices that meet specific criteria, and only these authorized devices can receive a unique decryption key.

[3] **Non-repudiation:** It states that the origin of data cannot be denied. It provides irrefutable proof that a specific person or entity sent or created particular information. It is crucial in digital transactions and communications as it prevents parties from disavowing their actions or commitments.

[4] **Collusion resistance:** Malicious users could pool their decryption keys to access protected data that none can open independently. The system should be robust enough to prevent such collaborative attacks.

[5] **Data integrity:** The system must rigorously safeguard the integrity, consistency, and accuracy of data stored on the blockchain from its creation to its final state. Once data is recorded on a blockchain cannot be altered or deleted without the network's consensus. It creates an unbreakable chain of records, making it virtually impossible to tamper with data.

[6] **Policy Privacy:** The access structure embedded within encrypted data should remain hidden from the blockchain and unauthorized users to ensure data privacy.

5.4.4 Security Model

Here, a security model structured as an interactive game between the two entities known as challenger $\mathfrak{C}$ and adversary $\mathfrak{A}$ is described, following the framework from Waters, B. [68]. The security process unfolds as follows:

Setup: The *Setup* algorithm is run to produce the parameters of the system. The $\mathfrak{C}$ uses the $Setup_{AA}$ algorithm to formulate the PK and the MSK. The challenger keeps the MSK and shares the public key PK with the $\mathfrak{A}$.

*Phase*1: The $\mathfrak{A}$ can make adaptive queries to the $\mathfrak{C}$ to obtain secret keys. Following these queries, $\mathfrak{C}$ executes the $KeyGen$ algorithm and provides the corresponding secret key to $\mathfrak{A}$.

Challenge: The adversary $\mathfrak{A}$ presents two messages, msg_0 and msg_1, of equal length,

along with a challenge access structure $\mathbb{A}^*$ that cannot be satisfied by any of the attributes inquired in $Phase1$. $\mathfrak{C}$ flips a coin $\zeta \in 0, 1$ and generates a ciphertext $CT^* \leftarrow Encrypt(GPP, PK, msg_\zeta, \mathbb{A}^*)$, which is then given to $\mathfrak{A}$.

*Phase*2**:** The $\mathfrak{A}$ continues to request secret keys from $\mathfrak{C}$ under the same conditions as in $Phase1$.

*Guess***:** The $\mathfrak{A}$ guesses the value of ζ as ζ'. The adversary's advantage is calculated as $|Pr[\zeta = \zeta'] - \frac{1}{2}|$.

Definition 1:*BloCPABE is secure if all polynomial-time adversaries have at most a negligible advantage in the above game.*

5.5 Concrete Construction

The section 5.5 thoroughly constructs the BloCPABE scheme. BloCPABE is constructed by taking inspiration and combining the idea of Waters, B. [68] and Yang et al. [42]. The scheme consists of algorithms such as system initialization ($Setup_{CA}, RegisterUser, Setup_{AA}$), $KeyGen, Encrypt, Decrypt, UKeyGen, Update_{SK}$, and $Update_{CT}$ (illustrated in Figure 5.3).

5.5.1 Proposed BloCPABE Scheme

Phase 1: System Initialization (Setup) : In this phase, algorithms such as $Setup_{CA}$, $RegisterUser$, and $Setup_{AA}$ are executed.

[1] ***Global Initialization (*$Setup_{CA}$*)*:** In this sub-algorithm, the CA utilizes a security parameter, denoted as λ, which dictates the length of the input group. The CA selects two multiplicative cyclic groups, $\mathbb{G}_1$ and $\mathbb{G}_T$, each of prime order p. Here, g serves as the generator of $\mathbb{G}_1$, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ represents a Bilinear map. CA selects a collusion-resistant hash function $\mathcal{H}$, which matches the binary string $\{0, 1\}^*$ to the random element in $\mathbb{G}_1$, expressed as $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{G}_1$. CA randomly selects two numbers, constructs MK, and publishes GPP to all the system members as shown in algorithm 5.1. Additionally, both AA and users are required to undergo registration with CA to acquire globally unique identifiers uid_k (for users) and aid (for AA) that validate their legal identities.

[2] ***RegisterUser:*** Here, DU and DO are referred to as users (with different roles) (subsection 5.4.2). It is assumed that CA runs the $RegisterUser$ algorithm within a

Algorithm 5.1 Global Initialization Algorithm

Input: Security Parameter λ

Output: Master Key MK, Global Public Parameter GPP

1: **PROCEDURE AT CERTIFICATE AUTHORITY**
2: Begin
3: $a, b \in_R \mathbb{Z}_p$ // Select random numbers
4: Compute g^a
5: Compute g^b
6: $MK \leftarrow (a, b)$
7: $GPP \leftarrow (g, g^a, g^b, \mathcal{H})$
8: End

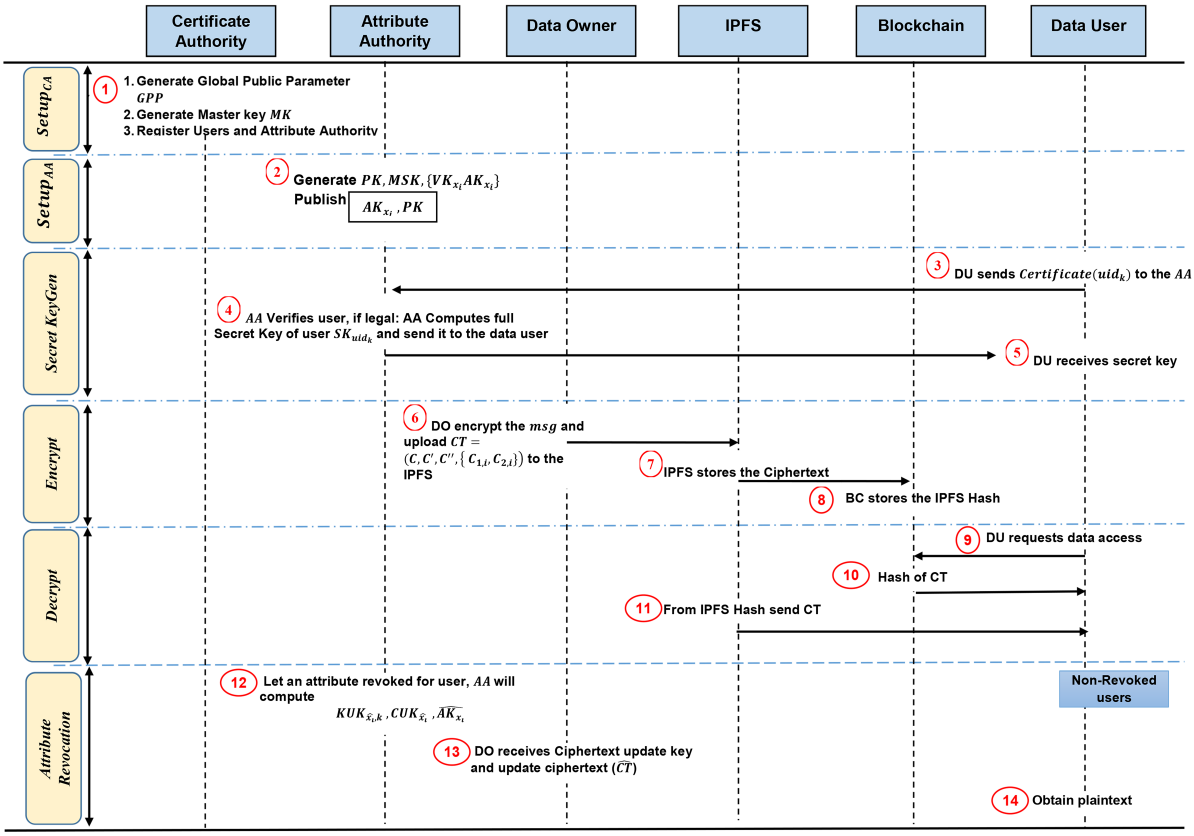


Figure 5.3: Illustration of the proposed scheme

secure environment, ensuring the confidentiality of all transmitted data. Upon system initialization, each user submits a registration request to the CA. The CA verifies the user's identity and subsequently assigns a uid_k and also creates $Certificate(uid_k)$ for uid_k. The identifier uid_k is a randomly generated value within the prime field $\mathbb{Z}_p$, i.e. $uid_k \in \mathbb{Z}_p$. After this, it randomly chooses the two numbers $\gamma_k, \gamma'_k \in_R \mathbb{Z}_p$ to generate the public/private key pair as $UPK_{uid_k} = g^{\gamma_k}, UPK'_{uid_k} = g^{\gamma'_k}$ and $USK_{uid_k} = \gamma_k, USK'_{uid_k} = \gamma'_k$.

[3] AA Setup ($Setup_{AA}$): This sub-algorithm is managed by an AA to generate the

MSK and PK. The AA within the system manages a universe of attributes $\mathcal{U}_{aid}$ and for each attribute $x_i \in_R \mathcal{U}_{aid}$, AA generates the public attribute key AK_{x_i} along with the version key VK_{x_i}. AA chooses two random exponents and compute PK and MSK as illustrated in algorithm 5.2.

Algorithm 5.2 Attribute Authority Setup Algorithm

Input: GPP, Authority aid attributes $\mathcal{U}_{aid}$

Output: Public Key PK, Master Secret Key MSK, Public Attribute Key of Attribute AK_x

PROCEDURE AT ATTRIBUTE AUTHORITY
Begin
$\alpha, y \in_R \mathbb{Z}_p$ // Select random numbers
$PK \leftarrow e(g,g)^{\alpha}$
$MSK \leftarrow (\alpha, y)$
for $x_i : (x_i \in \mathcal{U}_{aid})$ **do**
    $v_{x_i} \in_R \mathbb{Z}_p$
    $VK_{x_i} \leftarrow v_{x_i}$
    $AK_{x_i} \leftarrow (g^{v_{x_i}} \mathcal{H}(x_i))^y$
end for
End

For each attribute $x_i : (x_i \in \mathcal{U}_{aid})$, AA chooses version key as $VK_{x_i} = v_{x_i}$ and get the corresponding attribute key $AK_{x_i} = (g^{v_{x_i}} \mathcal{H}(x_i))^y$. Attributes are kept confidential and known only to AA, whereas PK and AK_{x_i} are published publicly with system entities.

Phase 2: Key generation (PHR user authorization): When the user seeks access to the data, it requests the secret key SK_{uid_k} from the AA as explained in the algorithm 5.3. AA assigns a set of attributes $S_k \in \mathcal{U}_{aid}$ to the legal user according to its identity and role in the system. Subsequently, uid_k gives $Certificate(uid_k)$ to the AA, and upon verifying the user's identity, the authority executes the $KeyGen$ algorithm.

Phase 3: Encryption Process (Outsourcing PHR on decentralized storage): Access to a data owner PHR is granted to authorized healthcare professionals, including physicians and nurses, upon satisfaction of specific conditions. Prior to outsourcing data to the decentralized storage, the PHR owner encrypts the data using an encryption algorithm. As discussed in subsection 5.4.2, it takes the public parameter, message, public keys of relevant authorities, and LSSS access structure. (M, ρ) is an access structure where M is a matrix with $l \times n$ elements, and ρ maps the rows of M into the attributes set. The DO subsequently selects a random column vector $\overrightarrow{v} = (s, v_2, v_3, ..., v_n)$, where $v_2, v_3, ..., v_n$ are applied to share s. DO computes $\lambda_i = M_i.\overrightarrow{v}$ for $\forall i \in [1, l]$, where M_i is a vector corresponding to the i^{th}

Algorithm 5.3 Key Generation For Users

Input: GPP, uid_k

Output: User's Secret Key SK_{uid_k}

PROCEDURE AT ATTRIBUTE AUTHORITY
Begin
if AA authenticates uid_k using $Certificate(uid_k)$ **then**
    Select t_k randomly, $t_k \in_R \mathbb{Z}_p$
    Compute $K_k \leftarrow g^{\alpha} g^{a\gamma_k} g^{bt_k}$
    Compute $K'_k \leftarrow g^{t_k}$
    Compute $K''_k \leftarrow g^{yt_k}$
    for all $x_i \in S_k$ **do**
        Compute $K_{x_i,k} \leftarrow g^{\gamma'_k y t_k} \cdot (g^{v_{x_i}} \mathcal{H}(x_i))^{y\gamma_k}$
    end for
    Compute $SK_{uid_k} \leftarrow \left\{K_k, K'_k, K''_k, \{K_{x_i,k}\}_{x_i \in S_k}\right\}$
else
    Exit
end if
End

row of matrix M. DO randomly choose $\tau_1, \tau_2, ..., \tau_l \in \mathbb{Z}_p$ and compute the CT as illustrated in 5.4.

Algorithm 5.4 Encryption Algorithm

Input: GPP, msg, policy

Output: Ciphertext CT

PROCEDURE AT DATA OWNER
Begin
Define Access Structure
Compute $C = msg \cdot e(g,g)^{\alpha s}$ //Compute Ciphertext Component
Compute $C' = g^s$
Compute $C'' = g^{bs}$
for $i = 1$ to l **do**
    Compute $C_{1,i} = g^{a\lambda_i} \cdot (g^{v_{\rho(i)}} \mathcal{H}(\rho(i)))^{y})^{-\tau_i}$, $\rho(i) \in \mathcal{U}_{aid}$
    Compute $C_{2,i} = g^{\tau_i}$
end for
$CT = \{C, C', C'', \{C_{1,i}, C_{2,i}\}_{i=1}^{l}\}$ //Construct Ciphertext
Return CT
End

Phase 4: Decryption Process (Accessing the PHR): To access the encrypted data, a DU submits a request to the decentralized storage. Those DUs that meet the access policy defined in the CT can decrypt the information. The DU runs the local decryption algorithm as briefly illustrated in subsection 5.4.2.

Case 1: If user attributes necessary of decryption $\notin \mathbb{A}$ then $\perp$.

Case 2: If user attributes necessary of decryption $\in \mathbb{A}$ then algorithm recovers the msg as: $I = \{i : p(i) \in \mathcal{U}_{aid}\} \subseteq [1, .., l]$ and compute the coefficient $\{w_i \in \mathbb{Z}_p\}_{i \in I}$ so that $\Sigma_{i \in I} w_i \lambda_i = s$. The decryption algorithm computes

$$\frac{e(C', K_k)e(C'', K'_k)^{-1}}{\prod_{i \in I}\Big(e(C_{1,i}, UPK_{uid_k})e(C_{2,i}, K_{\rho(i),k})e(C_{2,i}, K''^{USK'_{uid_k}}_k)^{-1}\Big)^{w_i N_A}}$$

to obtain the original message (msg), the user computes

$$msg = \frac{C}{e(g,g)^{\alpha s}}$$

Phase 5: Attribute Revocation: When an attribute $\widehat{x}_i$ is revoked for user uid'_k by AA, the system should implement a mechanism to invalidate the user's ability to decrypt ciphertexts containing that attribute. It prevents the revoked user from accessing PHRs protected by policies that require the revoked attribute. For users whose attributes remain valid, AA should update their secret key to maintain system security and integrity. The attribute revocation of our work is somewhat similar to the Yang et al. [42].

[1] *Update Key by AAs*: When $\widehat{x}_i$ is revoked from the user, the AA runs the $UKeyGen$ algorithm as illustrated in subsection 5.4.2. It firstly chooses a random versions key $v'_{\widehat{x}_i}$ and generate the attribute update key $AUK_{\widehat{x}_i} = y\Big(v'_{\widehat{x}_i} - v_{\widehat{x}_i}\Big)$, then AA compute the $KUK_{\widehat{x}_i,k}$ and $CUK_{\widehat{x}_i}$.

$KUK_{\widehat{x}_i,k} = g^{\gamma_k . y\Big(v'_{\widehat{x}_i} - v_{\widehat{x}_i}\Big)}$

$CUK_{\widehat{x}_i} = -\Big(y\Big(v'_{\widehat{x}_i} - v_{\widehat{x}_i}\Big)\Big)$

The AA update the attribute key of revoked attribute $\widehat{x}_i$ by computing

$\widehat{AK}_{\widehat{x}_i} = AK_{\widehat{x}_i} . g^{AUK_{\widehat{x}_i}}$

and send the message about $\widehat{x}_i$ updates to the data owner under the safe channel.

[2] *Secret Key Update for Non-revoked Users*: Upon receiving the $KUK_{\widehat{x}_i,k}$ from the AA, user will run the $Update_{SK}$ algorithm to update the secret key as

$\widehat{SK}_{uid_k} = \Big(\widehat{K_k} = K_k, \widehat{K'_k} = K'_k, \widehat{K''_k} = K''_k,$

$\widehat{K}_{\widehat{x}_i,k} = K_{\widehat{x}_i,k} . KUK_{\widehat{x}_i,k},$

$\forall x_i \in S_k, x_i \neq \widehat{x}_i : \widehat{K}_{x_i,k} = K_{x_i,k}\Big)$

The $KUK_{\widehat{x}_i,k}$ is associated with revoked uid'_k so that revoked users can easily be identified and differentiated. Consequently, revoked users cannot obtain the updated key.

[3] *Ciphertext Update by the Data Owner*: Upon receipt of the key $CUK_{\widehat{x}_i}$ from the AA, the owner will execute the $Update_{CT}$ algorithm to refresh the ciphertext. Initially, the data owner gathers the ciphertext components $(C_{1,i}, C_{2,i})$ that includes the attribute $\widehat{x}_i$. The updated ciphertext is then calculated as follows:

$\widehat{CT} = \Big(C = msg.e(g,g)^{\alpha s}, C' = g^s, C'' = g^{bs},$

$\forall i \in [1,l]:$

if $\rho(i) = \widehat{x}_i : \widehat{C}_{1,i} = C_{1,i}.(C_{2,i})^{CUK_{\widehat{x}_i}},$

if $\rho(i) \neq \widehat{x}_i : \widehat{C}_{1,i} = C_{1,i}, \widehat{C}_{2,i} = C_{2,i}\Big).$

5.5.2 Correctness Proof

Definition 2: *Our scheme is correct because the following equations are satisfied.*

Proof

$$\frac{e(C', K_k)e(C'', K'_k)^{-1}}{\prod_{i\in I}\Big(e(C_{1,i}, UPK_{uid_k})e(C_{2,i}, K_{\rho(i),k})e(C_{2,i}, K''^{USK'_{uid_k}}_k)^{-1}\Big)^{w_i N_A}}$$

$$= e\big(C', K_k\big)e\big(C'', K'_k\big)^{-1}$$

$$= e\big(g^s, g^\alpha g^{a\gamma_k} g^{bt_k}\big).e\big(g^{bs}, g^{t_k}\big)^{-1}$$

$$= e(g,g)^{s\alpha}.e(g,g)^{sa\gamma_k}.e(g,g)^{sbt_k}.e(g,g)^{-bst_k}$$

canceling

$$e(g,g)^{sbt_k}.e(g,g)^{-bst_k}$$

we get,

$$= e(g,g)^{sa\gamma_k}.e(g,g)^{s\alpha}$$

for each $i \in I$, assume $p(i) \in \mathcal{U}_{aid}$:

$$= \prod_{i\in I}\big(e\big(C_{1,i}, UPK_{uid_k}\big)e\big(C_{2,i}, K_{p(i),k}\big)e\big(C_{2,i}, K''^{USK'_{uid_k}}_k\big)^{-1}\big)^{w_i N_A}$$

simplifying the equation:

$$= e\big(g^{a\lambda_i}.((g^{v_{p(i)}}\mathcal{H}(p(i))^{y})^{-\tau_i}, g^{\gamma_k}\big).e\big(g^{\tau_i}, g^{\gamma'_k yt_k}.$$
$$(g^{v_{p(i)}}\mathcal{H}(p(i)))^{y\gamma_k}\big).e\big(g^{\tau_i}, g^{yt_k\gamma'_k}\big)^{-1}$$

$$= e\big(g,g\big)^{a\lambda_i\gamma_k}.e\big(g, g^{v_{p(i)}}\mathcal{H}(p(i))\big)^{-\gamma_k y\tau_i}.e\big(g,g\big)^{\tau_i\gamma'_k yt_k}.$$
$$e\big(g, g^{v_{p(i)}}\mathcal{H}(p(i))\big)^{\tau_i y\gamma_k}.e\big(g,g\big)^{-\tau_i yt_k\gamma'_k}$$
$$= e(g,g)^{a\lambda_i\gamma_k}$$

after we get,

$$\prod_{i\in I}\Big(e(g,g)^{a\lambda_i\gamma_k}\Big)^{w_i N_A} = e(g,g)^{sa\gamma_k}, \quad \Sigma_{i\in I} w_i\lambda_i = s,$$

then it computes,

$$\frac{e(g,g)^{sa\gamma_k}.e(g,g)^{\alpha s}}{e(g,g)^{sa\gamma_k}} = e(g,g)^{\alpha s}$$

to obtain the message

$$= \frac{C}{e(g,g)^{\alpha s}}$$
$$= \frac{msg.e(g,g)^{\alpha s}}{e(g,g)^{\alpha s}}$$
$$= msg$$

5.6 Security Analysis

Here in this section 5.6, a detailed security analysis of BloCPABE is discussed:

Theorem 1: *The proposed system model is protected against repudiation attack.*

Proof: By permanently recording data in a tamper-proof manner and verifying the identity of transaction participants through digital signatures, blockchain ensures that actions cannot be denied.

Theorem 2: *BloCPABE ensures the integrity and resistance to tampering.*

Proof: BloCPABE ensures the integrity and resistance to tampering with the encrypted data by utilizing the inherent properties of BT. Since the hash of encrypted data is stored on a blockchain, any attempts to tamper with or alter the data would require a consensus among the majority of nodes on the network, making it virtually impossible to manipulate the data without being detected. By its very design, blockchain is a decentralized and

immutable ledger that records transactions across a network of computers. Once data is entered into the blockchain, it is cryptographically linked to the previous blocks, forming a chain resistant to modification. Any attempt to alter a previous block would require recalculating and altering all subsequent blocks, which is computationally infeasible in a well-distributed network.

Theorem 3: *Our scheme protects and maintains the confidentiality of data, ensuring access is limited to authorized users only.*

Proof: BloCPABE facilitates robust, fine-grained access control, ensuring that sensitive data is securely delivered to authorized users while protecting it from unauthorized access. It guarantees that only users who satisfy the predefined access policy can access the encrypted data. The data owner encrypts and stores the data in decentralized storage while sharing the corresponding secret key with authorized users through a secure channel, preventing potential adversaries from accessing the plaintext.

Theorem 4: *The proposed framework is resistant to the collusion attack.*

Proof: In scenarios where multiple users collaborate, they might decrypt the ciphertext by combining the attributes. When malicious users collude, they may attempt to combine their attributes to decrypt the ciphertext. However, since the secret key is linked to each user's unique identifier *uid*, collusion between two users with different identifiers *uid* and uid' will not enable them to recover the factor $e(g,g)^s$. Consequently, such collusion will not allow these users to gain unauthorized privileges.

Theorem 5: *The scheme provides both backward and forward security.*

Proof: Forward security guarantees that once a user's decryption key (secret key) is revoked, they can no longer access newly encrypted data. Backward security, however, ensures that a newly generated decryption key cannot decrypt the encrypted ciphertext before the key's creation. It prevents a user from accessing sensitive information that was encrypted before they were granted access, thereby maintaining the confidentiality of historical data. The proposed scheme incorporates attribute revocation to guarantee both backward and forward security. Only users with valid attributes can update their secret keys to the latest version upon attribute revocation. Additionally, ciphertexts are updated using a ciphertext update key, preventing newly added users from accessing previously encrypted data.

Theorem 6: BloCPABE scheme is IND-CPA secure if the Decisional q-PBDHE assumption maintains, i.e., *No polynomial time adversary* $\mathfrak{A}$ *attempting to selectively break BloCPABE*

system with a challenge matrix of size $l^* \times n^*$ will succeed if the Decisional q-PBDHE assumption holds.

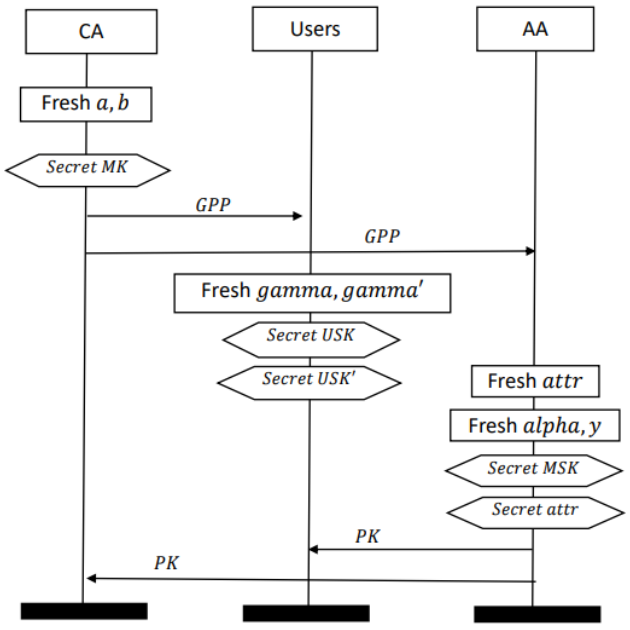
Proof: The details of the proof are referred to Waters, B. [68].

5.6.1 Formal Verification

To formally verify the protocol, BloCPABE has employed the Scyther Tool [125] as shown in Figure 5.4. Other than Scyther, there are numerous tools such as AVISPA [126], Proverif [127], and TAMARIN [128] that are used for the formal security analysis [6]. The Scyther tool analyzes a protocol described in SPDL and formulates security claims for each role specified in the protocol. It then verifies that these assertions are accurate; in other words, it checks the correctness of these claims. Scyther tool is designed to function with unlimited sessions, though this capability does not ensure indefinite correctness. In situations where this limitation applies, Scyther will indicate, "No attacks within bounds."' If the tool identifies no specific attacks or attack patterns, it will report 'No attacks.'

```
22 protocol BloCPABE(CA,Users,AA){
23 role CA{
24 claim(CA,Secret,MK);
25 send_1(CA,Users,GPP);
26 send_2(CA,AA,GPP);
27 recv_4(AA,CA,PK);
28 }
29
30 role Users{
31 claim(Users,Secret, USK);
32 claim(Users,Secret, USK');
33 recv_1(CA,Users,GPP);
34 recv_3(AA,Users,PK);
35 }
36 role AA{
37 claim(AA,Secret,MSK);
38 claim(AA,Secret,attr);
39 recv_2(CA,AA,GPP);
40 send_3(AA,Users,PK);
41 send_4(AA,CA,PK);
42 }
43 }
```

(a) Code snippet of setup phase



(b) Sequence diagram

Figure 5.4: Formal verification of system initialization phase

The implementation code of the entire BloCPABE protocol is very large, but to provide a brief understanding, a code snippet illustrating the system initialization of BloCPABE is given in Figure 5.4a. The code snippet showcases the exchange of messages between the

CA, AA, and users (where "users" refers to both DO and DU). Each entity exchanges data using the *send* and *recv* actions. The *send* action describes a participant in the protocol sending a message to another participant, while the *recv* action implies that a participant is prepared to accept an incoming message.

The BloCPABE protocol in Scyther contains five participants: CA, AA, DU, DO, and BC. Here, the Scyther tool also assesses claims such as $Weakagree$, which verifies that all parties involved are actively executing the protocol. $Nisynch$ ensures that data is handled consistently throughout the execution of the protocol. $Niagree$ guarantees that the sender believes the intended receiver actively participates in the one-way authentication protocol. *Alive* confirms the receiver's active status after the protocol's execution and verifies that they have initiated the expected response. Finally, $Commit$ confirms that the protocol has established a valid claim. The security is evaluated over five runs with adversarial

Table 5.2: Parameter setting in Scyther

Parameters	**Values**
Maximum number of runs	5
Matching type	Typed matching
Search Pruning	Find best attack
Maximum number of patterns per claim	10
Attack graph font size	14

parameters to determine the most suitable attack, with a maximum of 10 patterns per claim, as shown in Table 5.2.

Table 5.3: Scyther verification results

Claim				**Status**	**Comment**
BloCPABE	CA	BloCPABE,CA1	Secret keys(a,b)	Ok Verified	No attacks.
	Users	BloCPABE,Users1	Secret gamma	Ok Verified	No attacks.
		BloCPABE,Users2	Secret gamma'	Ok Verified	No attacks.
	AA	BloCPABE,AA1	Secret keys(alpha,y)	Ok Verified	No attacks.
		BloCPABE,AA2	Secret attr	Ok Verified	No attacks.
Done.					

In the setup phase (system initialization phase), attacks between the transmission from the CA to users, from the CA to the AA, and from the AA to the users are investi-

gated. The *Secret* claim ensures the protocol parameters remain confidential and protected from unauthorized observers or intruders. Scyther evaluates the following protocol claims: $claim(CA, Secret, MK)$ for the CA, $claim(Users, Secret, USK)$ and $claim(Users, Secret, USK')$ for the users, and $claim(AA, Secret, MSK)$ and $claim(AA, Secret, attr)$ for the AA. MK is the master key, and is defined in blocpabe.spdl as $macro\ MK = keys(a, b)$, where $keys$ is a function. Similarly, USK and USK' defined as $macro\ USK = gamma$ and $macro\ USK' = gamma'$. MSK is defined here as $macro\ MSK = keys(alpha, y)$. The sequence diagram Figure 5.4b illustrates that role CA performs a *send* action to send message/data to Users using $send\_1(CA, Users, GPP)$ and to AA using $send\_2(CA, AA, GPP)$. The Users claim the secrecy of USK, USK'. The role AA sends the message to the CA and the Users with $send\_3(AA, Users, PK)$ and $send\_4(AA, CA, PK)$ respectively, where PK is defined as $macro\ PK = map(g, g)power(alpha)$.

Scyther results : verify

Claim				Status		Comments
BloCPABE	CA	BloCPABE,CA1	Secret uid	Ok		No attacks within bounds.
		BloCPABE,CA2	Secret aid	Ok		No attacks within bounds.
		BloCPABE,CA3	Secret GMK	Ok	Verified	No attacks.
	AA	BloCPABE,AA1	Secret MSK	Ok		No attacks within bounds.
	DU	BloCPABE,DU1	Nisynch	Ok		No attacks within bounds.
		BloCPABE,DU2	Niagree	Ok		No attacks within bounds.
	DO	BloCPABE,DO2	Secret Cdata	Ok		No attacks within bounds.
		BloCPABE,DO3	Alive	Ok		No attacks within bounds.
		BloCPABE,DO4	Weakagree	Ok		No attacks within bounds.
		BloCPABE,DO5	Commit AA,ap,PK	Ok		No attacks within bounds.
		BloCPABE,DO6	Nisynch	Ok		No attacks within bounds.
		BloCPABE,DO7	Niagree	Ok		No attacks within bounds.
	BC	BloCPABE,BC1	Secret Cdata	Ok		No attacks within bounds.
		BloCPABE,BC2	Nisynch	Ok		No attacks within bounds.

Done.

Figure 5.5: Formal verification of BloCPABE using Scyther tool

The SPDL language is used to implement our suggested scheme, and the verification protocol results (Table 5.3) and Figure 5.5 have confirmed their correctness.

5.7 Performance Analysis

This section presents a theoretical analysis that includes a comparison of features or functionalities, along with an evaluation of computation (subsubsection 5.7.1.2), communication (subsubsection 5.7.1.3), and storage (subsubsection 5.7.1.4) overhead of relevant existing schemes. Additionally, it offers an experimental analysis of encryption and decryption times across varying numbers of attributes.

5.7.1 Theoretical Analysis

5.7.1.1 Features Comparison

A comparative analysis of several established CP-ABE schemes is conducted to evaluate the scalability of our scheme. The functionality of BloCPABE in comparison with the schemes are analyzed and described in [67], [42], [44], [129], [130], [18], [123], [69], [131] and [45], as detailed in Table 5.4.

As outlined in Table 5.4, an analysis is conducted based on several key aspects, including type, group type, and the access structure utilized. It also encompasses the level of renovation, whether at the attribute or user level, as well as support for key and ciphertext updates. Additionally, the analysis considers who is responsible for updating the ciphertext, the provision of non-repudiation, and protection against collusion attacks. We also assess whether the scheme ensures both forward and backward security, integrates BT for immutability and data integrity, and provides formal verification to confirm the absence of potential attack vectors. A "✓" indicates support for the technology, a "✗" suggests a lack of support, and "✖" denotes that it is not explicitly mentioned. This comparison focuses on various functional aspects critical for the practical deployment of these schemes in sensitive data-sharing environments. All the considered schemes are of the CP-ABE type and utilize the LSSS access structure, enhancing each scheme's expressiveness. Bethencourt et al. [67] scheme utilizes the tree access structure, and Riepel et al. [69] scheme uses the boolean formulae and MSP access structure. The scheme in [67], [42], [131], [44], [130], [18], [69], [123], [45] and our take prime order group, while [129] adopts composite order group.

In terms of revocation mechanisms, the schemes proposed by Yang et al. [42], Tu et al. [18], and our system support attribute-based revocation, while Wei et al. [44]scheme implements user-based revocation. Other schemes do not incorporate any form of revocation. Regarding key and ciphertext updates, the scheme [42], [44], [18], and our scheme facilitates

Table 5.4: Features comparison of proposed work with the existing solutions

Ref	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$	$P8$	$P9$	$P10$	$P11$	$P12$
[67]	ABE	Prime	Tree	✗	✗	✗	✗	✗	✖	✖	✗	✗
[42]	ABE	Prime	LSSS	Attribute	✓	✓	Server	✗	✓	✓	✗	✗
[131]	ABE	Prime	LSSS	✗	✗	✗	✗	✗	✗	✗	✗	✗
[44]	ABE	Prime	LSSS	User	✓	✓	Server	✗	✓	✗	✗	✗
[129]	ABE	Composite	LSSS	✗	✗	✗	✗	✗	✗	✗	✗	✗
[130]	ABE	Prime	LSSS	✗	✗	✗	✗	✗	✗	✓	✗	✗
[18]	ABE	Prime	LSSS	Attribute	✓	✓	Server	✗	✓	✓	✗	✗
[69]	ABE	Prime	Boolean Formulae and MSP	✗	✗	✗	✗	✗	✖	✖	✗	✗
[123]	ABE	Prime	LSSS	✗	✗	✗	✗	✓	✗	✗	✓	✗
[45]	ABE	Prime	LSSS	✗	✗	✗	✗	✓	✗	✗	✓	✗
BloCPABE	ABE	Prime	LSSS	Attribute	✓	✓	Owner	✓	✓	✓	✓	✓

✓= The scheme has the feature, ✗=absent, ✖= Unknown, The parameter $P1$ specifies the Type, $P2$: Order groups, $P3$: Access Structure, $P4$: Revocation level, $P5$: Key Update, $P6$:Ciphertext Update, $P7$:Ciphertext Updater, $P8$:Non-Repudiation, $P9$:Forward/Backward Security, $P10$:User Collusion Resistant, $P11$:Blockchain, $P12$:Formal Verification

both the key update and ciphertext update. The server handles the ciphertext update in [42], [44], and [18], while the owner is responsible for ciphertext updates in our scheme. When evaluating security characteristics, such as user collusion, forward/backward security, and non-repudiation, the schemes by [123], [45], and our scheme support non-repudiation and leverage BT for the secure storage of encrypted data, ensuring data integrity. The schemes [42], [44], [18], and our work supports the forward and backward security as well. Additionally, [42], [130], [18], and our scheme protects against user collusion attack. Also, our scheme is formally verified using the Scyther tool.

5.7.1.2 Computation Overhead

Before assessing the computational overhead, an examination of the execution time for each operation, as shown in Table 5.5, is conducted. The duration for each operation has been estimated utilizing the Charm framework [132] on a supersingular curve SS512. The analysis presented in Table 5.5 indicates that hash operations, exponentiation in $\mathbb{G}_1$ and $\mathbb{G}_T$, and pairing operations are significant contributors to the computational cost associated with ABE schemes.

Further, Table 5.6 illustrates the computational overhead across different schemes, highlighting the costs associated with key generation, encryption, and decryption processes. The following schemes are included in our evaluation: Yang et al. [42], Bethencourt et al. [67], Riepel et al. [69], Yamada et al. [131], Xu et al. [123], and Hou et al. [45].

Let T_E denote the time required for exponentiation operations in $\mathbb{G}_1$ and $\mathbb{G}_T$. Specifically, T_E is measured as 0.82201ms in $\mathbb{G}_1$ and 0.07321ms in $\mathbb{G}_T$. The $|S|$ represents the number

Table 5.5: Average execution time for each operation

Operation	**Time**
Multiplication Time in $\mathbb{Z}_p$	0.001129 ms
Exponentiation Time in $\mathbb{G}_1$	0.82201 ms
Exponentiation Time in $\mathbb{G}_T$	0.07321 ms
Multiplication Time in $\mathbb{G}_1$	0.00479 ms
Multiplication Time in $\mathbb{G}_T$	0.001039 ms
Inverse Operation Time in $\mathbb{G}_1$	0.000509 ms
Inverse Operation Time in $\mathbb{G}_T$	0.00323 ms
Time for Hash Operation	1.8171 ms
Time for Pairing Operation	0.537239 ms

of attributes possessed by user uid_k, which are utilized in constructing uid_k's secret key. Let T_P signify the time for pairing operations and H' for hash operations. The m denotes the number of attributes in the ciphertext. In schemes employing LSSS access structure, m corresponds to the number of rows in the LSSS matrix $M_{l\times n}$. The term I_e represents the minimum number of interior nodes required to satisfy an access structure, and $|I|$ indicates the number of attributes involved in the decryption of the ciphertext.

Table 5.6: Computation overhead of CP-ABE schemes

Schemes	**Key Generation**	**Encryption**	**Decryption**
Bethencourt et al.	$(2\|S\|+2)T_E + \|S\|H'$	$(2m+2)T_E + mH' + T_P$	$(2\|I\|+1)T_P + I_eT_E$
Yamada et al.	$(6\|S\|+2)T_E$	$(4m+2)T_E + T_P$	$(3\|I\|T_P)$
Yang et al.	$(3\|S\|+2)T_E$	$(5m+3)T_E + T_P$	$(4\|I\|T_P + \|I\|T_E)$
Riepel et al.	$(\|S\|+2)T_E+(\|S\|+1)H'$	$(2m)T_E + (m+1)H' + T_P$	$3T_P$
Xu et al.	$(6\|S\|+5)T_E$	$(6m+3)T_E + 2T_E$	$(4\|I\| + 1)T_P + \|I\|T_E + 4\|I\|T_E$
Hou et al.	$(8\|S\|+4)T_E$	$(3m+5)T_E$	$2T_E$
Our proposed	$(2\|S\|+3)T_E$	$(3m+3)T_E + T_P$	$(3\|I\|)T_P + \|I\|T_E$

This paper utilizes these time measurements (Table 5.5) in Table 5.6 to construct the computational times for key generation, encryption, and decryption with a varying number of attributes. The computation cost for the 100 number of attributes in Bethencourt et al. can be calculated as $(2|S|+2)T_E+|S|H'$, i.e., $(2\times100+2).0.82201+100\times1.8171 \approx 348.12ms$, let for Yamada et al. scheme keygen time is $(6 \times 100 + 2) \times 0.82201 = 494.85 \approx 498.26ms$. Our scheme's computation cost for keygen 100 attributes is $(2 \times 100 + 3) \times 0.82201 \approx 166.04ms$. Similar to the keygen, the computation cost for encryption for schemes such as Yang et al.,

for 100 number of attributes can be calculated as $(5 \times 100 + 3) \times 0.82201 + 0.537239 = 414 \approx 424.21ms$, including an additional overhead Δ associated with the encryption process.

The results depicted in Figure 5.9 and Figure 5.10 illustrate that the proposed scheme, compared to other schemes such as Bethencourt et al., Yamada et al., Yang et al., Riepel et al., Xu et al., and Hou et al., demonstrates superior efficiency in secret key generation. Similarly, our proposed scheme shows better results for encryption than Bethencourt et al., Yamada et al., Yang et al., Riepel et al., and Xu et al. The computation cost associated with the decryption (shown in Figure 5.11) of our scheme is comparatively higher than Riepel et al., Hou et al., Bethencourt et al., and Yamada et al. and lower than Yang et al. and Hu et al.(discussed in detail in subsection 5.7.2). Addressing the reduction of decryption costs while maintaining the security of the scheme is an area we plan to explore in future work.

5.7.1.3 Communication Overhead

The subsubsection 5.7.1.3 provides an overview of the communication overhead analysis, including the count of communicated messages and the overall cost incurred. Specifically, subsubsection 5.7.1.3 indicates that the number of public parameters for the schemes proposed by Bethencourt et al., Yamada et al., Yang et al., Riepel et al., Xu et al., Hou et al., and our own are 5, 7, 4, 3, 11, 8 and 4, respectively.

The communication overhead of various CP-ABE schemes is outlined in Table 5.7, which presents the public parameter, secret key, and ciphertext used in the schemes by Bethencourt et al., Yamada et al., Yang et al., Riepel et al., Xu et al., and Hou et al. These schemes utilize public parameters, as follows: Bethencourt et al., includes $\{\mathbb{G}_0, g, h, f, e(g,g)^{\alpha}\}$, Yamada et al. employ $\{g, H, U, V, V', W, e(g,g)^{\alpha}\}$, Yang et al. use $\{g, g^a, g^b, \mathcal{H}\}$, Riepel et al. incorporates $\{\mathcal{G}, \mathcal{H}, e(g_1, g_2)^{\alpha}\}$, Xu et al. use $\{g, g_2, {g_2}^{\eta}, c, u, w, h, v, Y, Q, H\}$ and Hou et al. includes $\{D, g, w, v, u, h, e(g,g)^{\alpha}, H\}$. The number of messages in the secret key for each scheme is 3, 6, 3, 3, 7, and 7, respectively. These are as follows: Bethencourt et al. includes D, D_j, D'_j, Yamada et al. incorporates $\{D_1, D_2, \{K_{i,1}, K_{i,2}, K'_{i,1}, K'_{i,2}\}\}$, Yang et al. uses $\{K_{uid,aid}, K'_{uid,aid}, \{K_{x_{aid},uid}\}\}$, Riepel employs $\{sk_1, \{sk_2\}, sk_3\}$, Xu et al. use $\{K_0, K_1, \{K_{2,i}, K_{3,i}, K_{4,i}, K_{5,i}, K_{6,i}\}_{i \in I_S}\}$ and Hou et al. $\{sk_1, sk_2, sk'_2, \{sk_{i,3}, sk_{i,4}, sk'_{i,3}, sk'_{i,4}\}_{i \in [1,k]}\}$. Similarly, in the ciphertext, the communicated messages are as follows: Bethencourt et al. includes $\{\widetilde{C}, C, \{C_y, C'_y\}\}$, Yamada et al. use $\{C_0, C_1, \{C_{i,1}, C_{i,2}, C_{i,3}\}\}$, Yang et al. employ $\{C, C', C'', \{C_i, C'_i, D_i, D'_i\}\}$, Riepel et al. includes $\{ct_1, \{ct_{2,j}\}, \{ct_{3,i}\}\}$, Xu et al. has $\{C_0, C'_0, C_1, \{C_{2,j}, C_{3,j}, C_{4,j}, C_{5,j}, C_{6,j}, C_{7,j}\}_{j \in [1,l]}\}$, and Hou et al. includes $\{C_0, C_1, \{C_{i,2}, C_{i,3},$

Table 5.7: Total communication overhead of CP-ABE schemes

Schemes	Public Parameter	Secret Key	Ciphertext	Total Overhead
Bethencourt et al.	5	3	4	12
Yamada et al.	7	6	5	18
Yang et al.	4	3	7	14
Riepel et al.	3	3	3	9
Xu et al.	11	7	9	27
Hou et al.	8	7	6	21
Our proposed	4	4	5	13

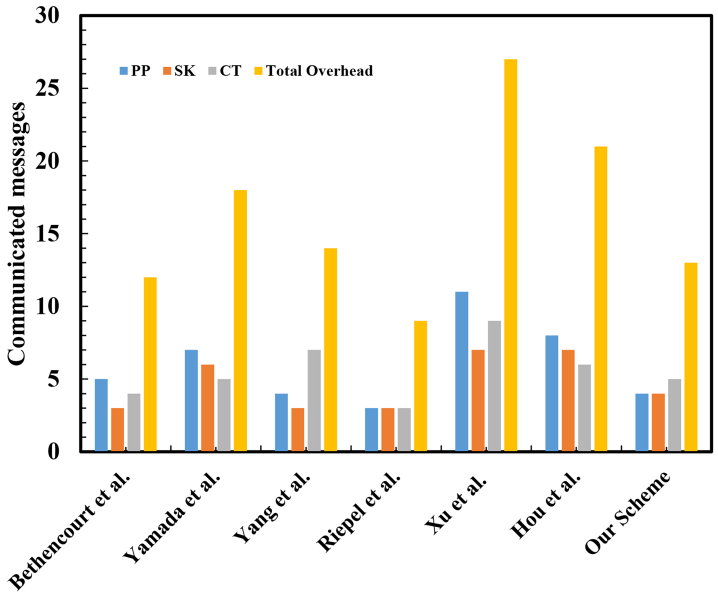


Figure 5.6: Communication overhead of various schemes

$C_{i,4}\}_{i\in[1,l]}, C_5\}$.

As elaborated in section 5.5, our scheme utilizes PP $g, g^a, g^b, \mathcal{H}$. Furthermore, SK in our scheme comprises the set $\{K, K', K'', \{K_{x_i}\}\}$, as detailed in Figure 5.6. The total number of communicated messages in ciphertexts is 5, represented as $CT = \{C, C', C'', \{C_{1,i}, C_{2,i}\}\}$. When compared to the schemes by Yamada et al. and Yang et al., the communication overhead of our scheme remains relatively affordable, as illustrated in Figure 5.6.

5.7.1.4 Storage Overhead

The storage overhead, as shown in Figure 5.7, Figure 5.8, is computed and compared with the schemes proposed by Bethencourt et al., Yamada et al., Yang et al., Riepel et al., Xu et al., Hou et al., and our proposed scheme. This analysis evaluates the secret key and

ciphertext sizes for different CP-ABE schemes as shown in Table 5.8. Let $|L_{\mathbb{G}_1}|$ represent the bit size of an element in the source group $\mathbb{G}_1$, and $|L_{\mathbb{G}_T}|$ represent the bit size of an element in the target group $\mathbb{G}_T$. These schemes are based on the symmetric pairing SS512, where in the elliptic curve SS512, the sizes of the fields corresponding to $|\mathbb{Z}_p|$, and $|L_{\mathbb{G}_1}|$ are 20 bytes, and 64 bytes respectively. These field sizes are applied to calculate the size of ciphertext and secret key in Figure 5.7, Figure 5.8, and the results are depicted in subsubsection 5.7.1.4.

Table 5.8: Storage overhead comparison of proposed work with the existing schemes

Schemes	**Secret Key Size**	**Ciphertext Size**
Bethencourt et al.	$(2\|S\|+1)\|L_{\mathbb{G}_1}\|$	$(2m+1)\|L_{\mathbb{G}_1}\|+1\|L_{\mathbb{G}_T}\|$
Yamada et al.	$(4\|S\|+2)\|L_{\mathbb{G}_1}\|$	$(3m+1)L_{\mathbb{G}_1}+1\|L_{\mathbb{G}_T}\|$
Yang et al.	$(\|S\|+2)\|L_{\mathbb{G}_1}\|$	$(4m+2)\|L_{\mathbb{G}_1}\|+1\|L_{\mathbb{G}_T}\|$
Riepel et al.	$(\|S\|+1)\|L_{\mathbb{G}_1}\|$	$(m+1)\|L_{\mathbb{G}_1}\|+1\|L_{\mathbb{G}_T}\|$
Xu et al.	$(5\|S\|+2)\|L_{\mathbb{G}_1}\|$	$(6m+2)\|L_{\mathbb{G}_1}\|+1\|L_{\mathbb{G}_\mathbb{T}}\|$
Hou et al.	$(4\|S\|+3)\|L_{\mathbb{G}_1}\|$	$(3m+2)\|L_{\mathbb{G}_1}\|+1\|L_{\mathbb{G}_\mathbb{T}}\|$
Our proposed	$(\|S\|+3)\|L_{\mathbb{G}_1}\|$	$(2m+3)\|L_{\mathbb{G}_1}\|+1\|L_{\mathbb{G}_T}\|$

It can be seen from Figure 5.7 that our scheme results in significantly lower storage overhead compared to the Yamada et al. $(4|S|+2)|L_{\mathbb{G}_1}|$, Bethencourt et al. schemes $(2|S|+1)|L_{\mathbb{G}_1}|$, Xu et al. $(5|S|+2)|L_{\mathbb{G}_1}|$ and Hou et al. $(4|S|+3)|L_{\mathbb{G}_1}|$. Additionally, the difference in storage overhead between the Riepel et al. $(|S|+1)|L_{\mathbb{G}_1}|$, Yang et al. $(|S|+2)|L_{\mathbb{G}_1}|$ and our proposed scheme $(|S|+3)|L_{\mathbb{G}_1}|$ is minimal. Regarding ciphertext storage, Riepel has the minimum cost, expressed as $(m+1)|L_{\mathbb{G}_1}|+1|L_{\mathbb{G}_T}|$. The Bethencourt et al. scheme also has a relatively low overhead at $(2m+1)|L_{\mathbb{G}_1}|+1|L_{\mathbb{G}_T}|$, which is close to the size of our ciphertext $(2m+3)|L_{\mathbb{G}_1}|+1|L_{\mathbb{G}_T}|$. Compared to Yang et al., Yamada et al., Xu et al., and Hou et al., our scheme demonstrates lower storage overhead (Figure 5.8).

5.7.2 Experimental Analysis

This section outlines a simulation-based experimental study to compare the proposed scheme with state-of-the-art algorithms in the field. The experiment is conducted across three common aspects: key generation, data encryption, and data decryption. All the experiments are carried out on a personal computer equipped with an AMD Ryzen 5 5500U processor featuring Radeon Graphics, 16GB of RAM, and an SSD 512 GB [133], running Ubuntu 22.04 LTS as the operating system. Our algorithms are implemented in Python 3.10.0 utilizing the "Charm-Crypto" [132] framework on a 512-bit supersingular elliptic curve (SS512) with

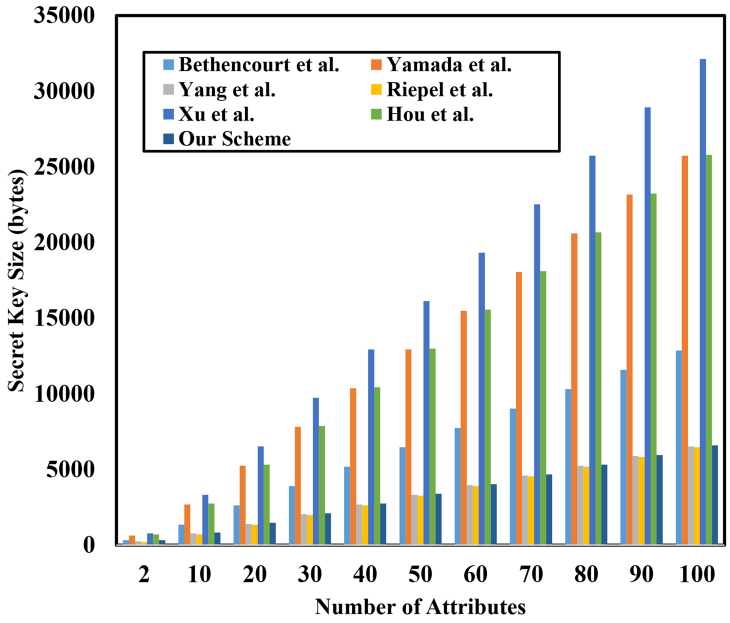


Figure 5.7: Key size

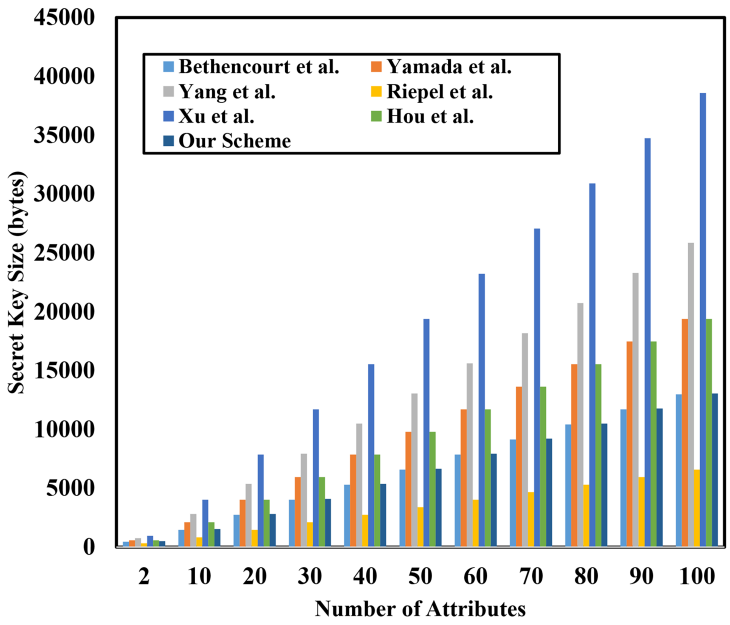


Figure 5.8: Ciphertext size

160-bit order.

The time efficiency of the CP-ABE schemes, as demonstrated in Figure 5.9, Figure 5.10 and Figure 5.11, is evaluated with respect to key generation, encryption, and data decryption. Note that some schemes do not address attribute revocation; therefore, only the common aspects are compared in this analysis. In the key generation stage, the time consumption of the key generation algorithm by varying the number of attributes is evaluated, as illustrated

in Figure 5.9.

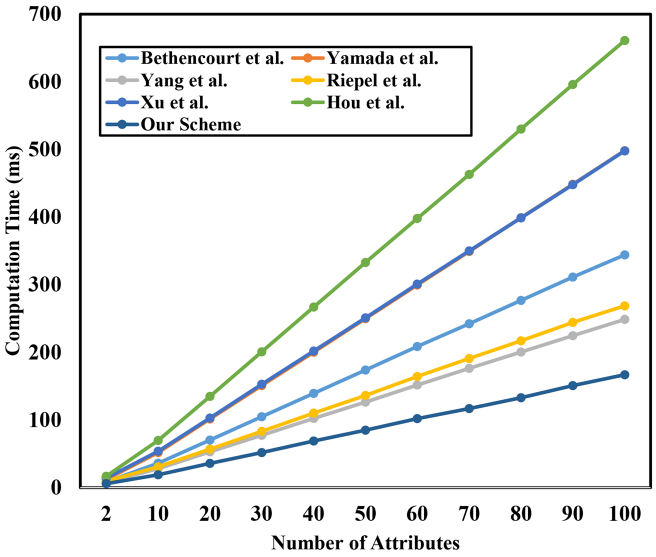


Figure 5.9: Key generation

As the number of attributes increases, a corresponding rise in key generation time is observed for all seven schemes. While Yang et al.'s scheme is designed for multi-authority ABE, we evaluated it in a single-authority for computational time comparison. Our proposed scheme demonstrates a notable advantage in key generation time compared to Bethencourt et al., Yamada et al., Yang et al., Riepel et al., Xu et al., and Hou et al.. When evaluating with 10 attributes, the key generation times for Bethencourt et al., Yamada et al., Yang et al., Riepel et al., Xu et al., Hou et al., and our scheme are 36.22ms, 51.95ms, 28.21ms, 30.78ms, 53.4ms, 69.3ms, and 16.24ms, respectively. Similarly, when evaluating 100 attributes, the times are 348.12ms, 498.26ms, 248.78ms, 268.84ms, 497.2ms, 659.1ms, and 167ms, respectively.

In order to guarantee consistency in the encryption and decryption process, each ciphertext is produced based on "AND" access policies, such as $ATTR_1$ AND $ATTR_2$ AND $ATTR_3$ AND ... AND $ATTR_N$, where N ranges from 2 to 100. Also, to ensure the reliability of execution time data, each experiment is conducted 20 times. The average of these trials is used to calculate the final execution time. As depicted in Figure 5.10, the encryption algorithm's execution times for access policies with 2 to 100 attributes are measured across seven schemes: Bethencourt et al., Yamada et al., Yang et al., Riepel et al., Xu et al., Hou et al., and our proposed scheme.

The results indicate that encryption under these policies consumed 36.69ms, 35.25ms, 44.98ms, 40.03ms, 50.9 ms, 28.5ms and 29.33ms, respectively, for 10 attributes; and 352.86ms, 340.32ms, 424.21ms, 363.48ms, 498.2ms, 257 ms and 256.87ms for 100 attributes. Our pro-

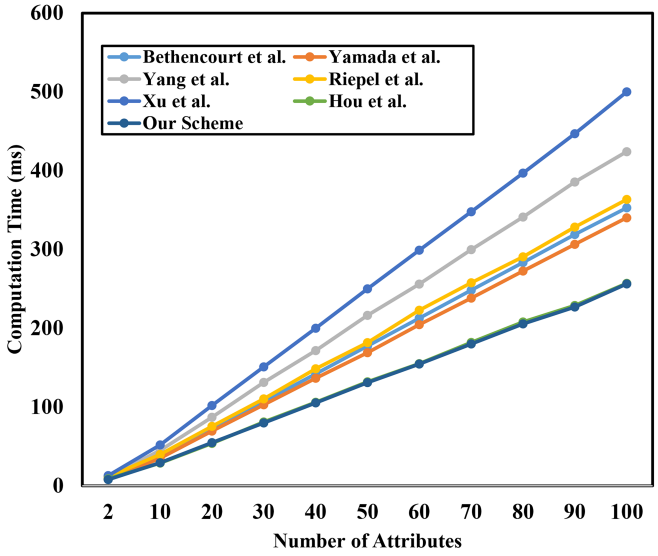


Figure 5.10: Encryption

posed encryption scheme demonstrates a notable improvement in computational efficiency compared to existing approaches. While there is a marginal performance gap between our scheme and Hou et al.'s, our solution exhibits better encryption performance. As depicted in

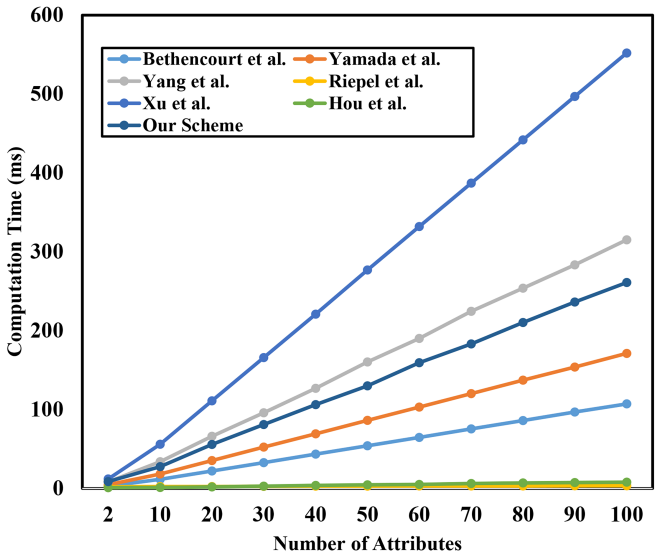


Figure 5.11: Decryption

Figure 5.11, the execution time of the decryption algorithm for access policies with 2 to 100 attributes is measured across seven schemes: Bethencourt et al., Yamada et al., Yang et al., Riepel et al., Xu et al., Hou et al., and our proposed scheme. The results indicate that decryption under these policies consumed 11.64ms, 18.35ms, 33.93ms, 2.37ms, 55.3ms, 1.2ms and 27.79ms, respectively, for 10 attributes; and 107.16ms, 171.23ms, 315.20ms, 3.50ms, 550.1 ms, 7.9 ms and 130.20ms for 100 attributes. The decryption time results presented in

Figure 5.11 indicate that our scheme takes more time than Bethencourt et al., Yamada et al., Hou et al., and Riepel et al. This additional computational overhead arises from pairing operations and secret key processing tasks associated with each attribute the user possesses, which increases with the number of attributes stated in the access policy. Through rigorous theoretical and empirical analysis, our proposed scheme demonstrates notable functionality and computational efficiency advantages compared to existing approaches. The detailed simulation of our proposed CP-ABE scheme with blockchain is analyzed in subsection 5.7.3.

5.7.3 Blockchain Simulation

To evaluate the effectiveness of the proposed model, a prototype illustrated in Figure 5.2 and Figure 5.3 is tested on a system specification given in subsection 5.7.2. We developed the smart contract using Solidity and tested it on RemixIDE. For blockchain simulation, BloCPABE utilized Ethereum on Ganache Truffle Suite [134], leveraging Web3.js as the communication interface. In this simulation, we assessed our CP-ABE scheme when integrated with IPFS 0.20.0 (IPFS) and the Ethereum blockchain. The primary objective is to evaluate the efficiency of the ABE scheme by leveraging IPFS for decentralized storage and Ethereum for recording metadata. Initially, a smart contract is deployed on a

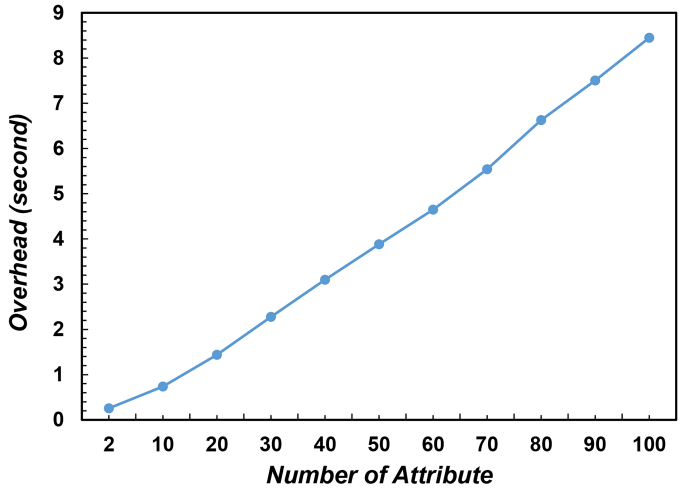


Figure 5.12: Overhead analysis with blockchain interaction

local Ethereum blockchain (Ganache) and is designed to store IPFS hashes associated with encrypted data. Encrypted content is serialized and uploaded to IPFS, ensuring a decentralized and immutable storage solution. Figure 5.12 details the overhead associated with the entire process, including the encryption time, the outsourcing of encrypted data to IPFS, and the recording of the hash on the blockchain. The time taken increases with the number of attributes. With 2 attributes, the observed overhead is 0.256 seconds; with 50 attributes,

it rises to 3.882 seconds; and with 100 attributes, it grows to 8.447 seconds. This increase in processing time highlights the challenges that arise as the array of attributes increases; the complexity and time required for encryption, data outsourcing to IPFS, and blockchain interactions become significant.

5.8 Summary

The security, privacy, and access control challenges in mHealth systems are addressed with the proposal of the BloCPABE scheme. Additionally, the system facilitates the efficient revocation of attributes with minimal computational latency, guaranteeing that user access rights are promptly updated in response to attribute modifications. The results demonstrate that the BloCPABE is computationally more efficient than existing ABE methods. In particular, our scheme observed a reduction in key generation costs by $\approx$ 33% to 74% and encryption costs a maximum of $\approx$ 49%, underscoring the efficiency enhancements of the approach. The BloCPABE scheme is designed to resolve critical concerns in healthcare data sharing by ensuring user privacy, forward/backward security, and resistance to collusion attacks. The solution further enhances security by guaranteeing the immutability and tamper-proof integrity of outsourced data through the integration of blockchain. These findings reveal that the BloCPABE scheme is highly secure and privacy-preserving, rendering it appropriate for real-time, fine-grained access control in mHealth systems. This work has substantial implications, as it offers a solution for enhancing the security and privacy of patient health records and enabling the efficient exchange of sensitive health data. Through theoretical complexity analysis and experimental experiments, the efficiency and feasibility of the scheme are verified.

In the next chapter, we introduce a smart contract vulnerability detection method named SmarConTesT. This approach employs a multi-label classification technique to identify multiple vulnerabilities within a single smart contract simultaneously. Leveraging stacking generalization, SmarConTesT integrates predictions from multiple base classifiers to enhance detection accuracy and robustness. The chapter details the design, implementation, and evaluation of the method, demonstrating its effectiveness in identifying multiple vulnerabilities in Ethereum smart contract.

Chapter 6

Smart Contract Vulnerability Detection

This chapter introduces SmarConTest, an effective approach for addressing smart contract vulnerabilities, utilizing multi-label categorization on a balanced dataset. This approach provides a comprehensive assessment of the security framework for smart contracts. Create a robust framework to detect different vulnerabilities in a single smart contract. Our experiments on the Slither-Audited Smart Contracts dataset show an average accuracy of ≈ 95.05% in identifying access control issues, arithmetic errors, reentrancy attacks, unchecked calls, and others. SmarConTest outperforms leading techniques in the field. After providing bytecode, SmarConTest detects smart contract vulnerability in about 0.11 seconds.*

6.1 Introduction

Addressing vulnerabilities in smart contracts is essential in the blockchain technology domain, particularly as decentralized applications proliferate across several sectors. The principles of immutability and tamper-resistance support the "code is law" theory, signifying that once conditions are set and disclosed, they remain unalterable [135]. Smart contracts, as self-executing agreements with encoded terms, offer benefits such as transparency and efficiency.

Attackers are increasingly targeting smart contracts for various substantial reasons. Primarily, many smart contracts on the Ethereum network facilitate monetary transactions, making them appealing targets. Secondly, once a susceptible smart contract is deployed on the blockchain, it becomes immutable, making it exposed to exploitation [136]. Ulti-

*The research work covered in this chapter has been published/accepted/communicated in: Anita Thakur, Virender Ranga and Ritu Agarwal, "SmarConTest: An Efficient Smart Contract Vulnerability Detection Method Using Machine Learning,"—communicated
&
Anita Thakur,Virender Ranga and Ritu Agarwal, "A Review On Smart Contract Vulnerability Detection Using Deep Learning",—communicated

mately, the lack of standardized assessment metrics for evaluating the quality and security of smart contracts heightens their vulnerability to attacks [137]. Sometimes, vulnerabilities go undetected post-deployment, hampering the implementation of remedies.

Table 6.1: Vulnerability detection tools using deep learning methods

Tool	Technique	Model	Vulnerability	Dataset Used/Created
Vulnsense [33]	DL	BERT, BiLSTM, GNN	Reentrancy, Arithmetic	https://smartbugs.github.io https://github.com/DependableSystemsLab/SolidiFI
CodeNet [38]	DL	CNN	Timestamp Dependency, Tx.origin,Unchecked Low-Level Calls, Reentrancy	https://github.com/DependableSystemsLab/SolidiFI https://github.com/smartbugs/smartbugs-wild
Liu et al. [32]	DL	GNN+Expert knowledge	Infinite loop,TOD,Reentrancy	https://github.com/ethereum/go-ethereum https://github.com/vntchain/go-vnt
ContractGNN [138]	ML	VSG+GNN	Timestamp dependency, Integer Overflow, Reentrancy	https://github.com/Wang-Yi-chen/ContractGNN
DL4SC [37]	ML	CNN+Transformer	Arithmetic, Reentrancy, Timestamp dependence	https://github.com/jacknichao/Automatic_Identification_of_Crash_Smart_Contracts https://github.com/LaoSuanNaii/DL4SC https://goo.gl/UAUpK5/
Deng et al. [34]	DL	Transformer+CNN	TOD, Locked-ether, Arithmetic, Reentrancy	https://github.com/sujeetc/ScrawlD
DeepFusion [35]	DL	Bi-LSTM+Attention	Self-destruct, Integer overflow, Tx.origin, Timestamp, Reentrancy	https://github.com/Tourneso/DeepFusion
Li et al. [139]	DL	Bi-LSTM, GCN	Timestamp dependency, Integer overflow, tx.origin, reentrancy	https://smartbugs.github.io https://github.com/DependableSystemsLab/SolidiFI
GRATDet [140]	DL	Transformer–GP	Reentrancy	https://github.com/wuhongjun15/Peculiar
Qian et al. [36]	DL	BERT model, GAT network	Delegatecall, Timestamp dependence, integer overflow/underflow, Reentrancy	https://github.com/Messi-Q/Cross-Modality-Bug-Detection
Narayana et al. [137]	DL	ANN, Auto-encoder	Tx.origin, Reentrancy, DoS	https://github.com/klngithubsairam/Research
DA-GNN [141]	DL	GNN+Attention	Self-destruct, TOD, integer overflow	https://github.com/ZZXLX/contract_hex/tree/master
Jain et al. [142]	DL	Bi-GRU, Text-CNN	Self-destruct, Low-level calls, Block timestamp, Inline assembly, Check effect, TOD, Timestamp dep, Mishandled exceptions, Call Stack, Overflow, Underflow, tx.origin, Reentrancy	https://github.com/jianwei76/SoliAudit https://github.com/smartbugs/smartbugs-wild
Zhang et al. [143]	DL	Bi-GRU+Attention, SVM	Reentrancy	https://github.com/wobulijie10086/deection/tree/master
Zhang et al. [144]	DL	...	Reentrancy	https://smartbugs.github.io
Tang et al. [31]	DL	Optimized-CNN, Optimized-LSTM, and Optimized-CodeBERT	Tx.origin, Timestamp dependence, Reentrancy, Unhandled-exception	https://huggingface.co/datasets/mwritescode/slither-audited-smart-contracts https://github.com/smartbugs/smartbugs-wild

Many ML and DL detection approaches employ different representations of contract code to discover potential vulnerabilities, as indicated in Table 6.1. Typical input formats for these tools include ASTs, opcodes, raw Solidity code, and CFG. DL4SC [37] and Contract-GNN [138] exemplify machine learning-based methodologies utilized for the identification of vulnerabilities, including timestamp dependency, integer overflow, reentrancy, and arithmetic errors. Similarly, [31], [144], [143], [142], and DA-GNN [141] employ DL methodologies

for the identification of vulnerabilities in smart contracts.

6.2 Motivation

A vulnerable smart contract becomes exploitable once deployed on the blockchain because it becomes immutable [136]. It is even easier to misuse smart contracts since there are no standardized measures for evaluating their quality and security [137]. Unfortunately, vulnerabilities can sometimes go undiscovered after deployment, which makes it difficult to provide fixes. The importance of finding weaknesses in smart contracts is highlighted in the paper [21]. Because they are based on expert-specified criteria, most vulnerability detection technologies now in use are slow and can't scale well. Methods like symbolic execution, fuzzing detection, formal verification, intermediate representation, and DL can be used to find smart contract vulnerabilities. Because even a small vulnerability in the smart contract could lead to huge losses, testing them thoroughly before deploying them on the blockchain is crucial to ensure they are bug-free.

6.3 Research Contributions

This paper introduces SmarConTest, an ML-based method to identify vulnerabilities in smart contracts. SmarConTest employs a stacking classifier, an efficient ensemble learning technique, to simultaneously identify multiple vulnerabilities through multi-label classification. The stacking classifier combines several fundamental models, including random forest, decision tree, and XGBoost, to clarify diverse patterns and relationships in the data. The basic models are then combined using logistic regression as a meta-model, which learns to optimally integrate the outputs of the different classifiers to produce more accurate predictions.

The methodology is applied to widely used smart contracts written in Solidity, effectively identifying vulnerabilities such as reentrancy, access control issues, unchecked calls, arithmetic errors, and other critical vulnerabilities . This study presents the following contributions:

[1] We introduce an ensemble learning-based approach to smart contract vulnerability detection with multiple labels. This method makes it possible to find multiple vulnerabilities in a single contract.

[2] The research utilizes an automated Opcode extraction technique to obtain Opcodes

from smart contracts and subsequently transforms them into vectorized embeddings using the TF-IDF vectorizer.

[3] The efficacy of the model is assessed and compared employing the Slither Audited Smart Contract Dataset [145]. The experimental results demonstrate that the model achieves an accuracy of 95.05

[4] SmarConTest recognizes all specified vulnerabilities in approximately 0.11 seconds.

6.4 Technical Background

This section presents an broad summary of various prevalent vulnerabilities in smart contracts, and ML models/approaches utilized for detection, and performance measures to evaluate the efficacy of our proposed framework.

6.4.1 Source code, Bytecode and Opcode

A smart contract functions as a self-executing digital agreement, encoded directly into the programming language and generally implemented on a blockchain platform. The execution process initiates upon the fulfillment of specified conditions, activating the contract to perform the stipulated actions without the involvement of intermediaries autonomously [37]. Compiling a smart contract consists of multiple steps that transform high-level code, usually authored in a language such as Solidity, into executable bytecode suitable for deployment on the blockchain. The developer begins by writing the contract code, followed by a verification process to identify syntax errors. Subsequently, a compiler converts this code into bytecode, a low-level format comprehensible to the EVM. In addition to the bytecode, essential metadata, including the contract's ABI, is generated for deployment. Bytecode is ultimately deployed to the blockchain, where it is assigned a unique address. It enables users to interact with the contract via transactions, as demonstrated in Figure 6.1.

In the EVM, bytecode comprises opcodes, each denoting a particular instruction that the EVM interprets and executes. These opcodes pertain to many processes, including arithmetic, data storage, and control flow. The opcode POP, denoted by the hexadecimal value $0x50$, eliminates the top element from the stack, whereas the opcode $PUSH1$, indicated by $0x60$, inserts 1 byte of data onto the stack, with the subsequent byte representing the value being pushed as illustrated in Figure 6.2. The direct correspondence between bytecode and opcodes (shown in Table 6.2) streamlines the examination of smart contracts, enhancing the efficiency of vulnerability detection.

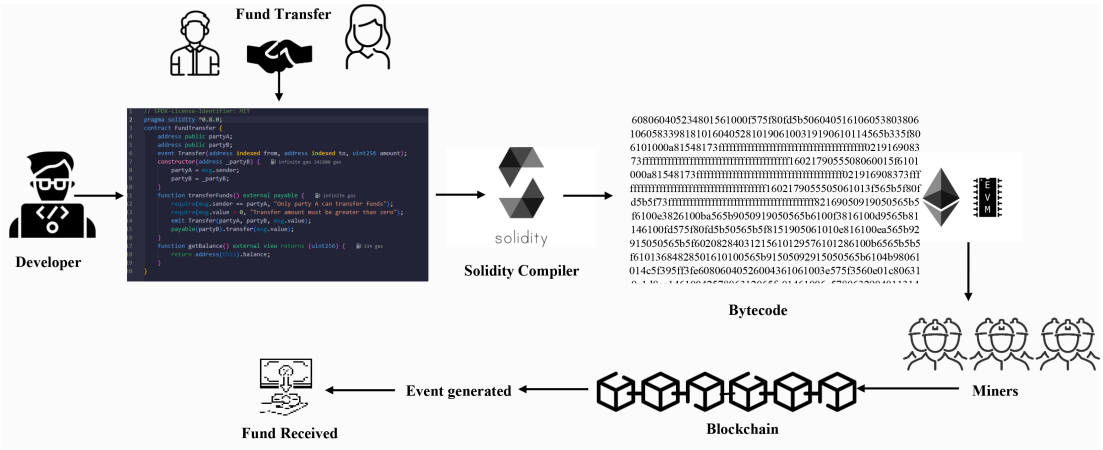


Figure 6.1: Execution of smart contract

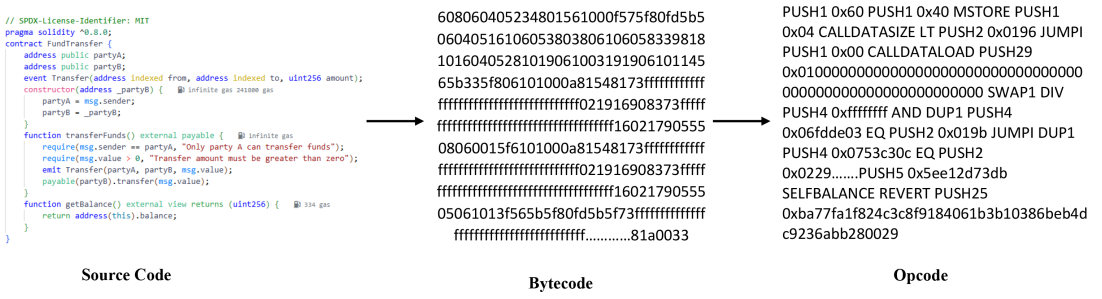


Figure 6.2: Source code, Bytecode and Opcode

6.4.2 Smart Contract Vulnerability

Vulnerabilities in smart contracts are weaknesses that bad actors can exploit in coding decentralized systems. Implementing comprehensive security and audit procedures throughout development is crucial to prevent unauthorized access, fund loss, or contract behavior manipulation caused by these vulnerabilities. The vulnerabilities listed in Figure 6.3 [21] and [146] are rather common. According to the SWC, smart contracts can be systematically categorized based on the vulnerabilities and vulnerabilities they may include.

A total of 49 vulnerabilities are given four different severity levels—critical, high, medium, and low—by Zhang et al. [147]. Our research focuses on weaknesses—reentrancy, arithmetic, unchecked calls, access control, and others—with some vulnerabilities rated as significant and others as high. The following is an explanation of the vulnerabilities:

[1] ***Reentrancy :***A reentrancy attack is a vulnerability in a smart contract that allows an attacker to modify the contract's state or drain funds by repeatedly calling a function before its previous execution completes. To exploit the unchanging state,

Table 6.2: Opcode and description from yellow paper

EVM code	Original opcode	Description
0x00, 0x01, 0x02..., 0x09, 0x0a, 0x0b	STOP ADD MUL SUB DIV SDIV MOD SMOD ADDMOD MULMOD EXP SIGNEXTEND	Stop and Arithmetic operations
0x0c ... 0x0f	unused	unused
0x10, 0x11,...,0x1c, 0x1d	LT GT SLT SGT EQ ISZERO AND OR XOR NOT BYTE SHL SHR SAR	Comparision, bitwise operation
0x20	KECCAK256	Computing hash
0x21,...,0x2f	unused	unused
0x30, 0x31,...,0x3e, 0x3f	ADDRESS BALANCE ORIGIN CALLER-CALLVALUE CALLDATALOAD CALL-DATASIZE CALLDATACOPY CODESIZE CODECOPY GASPRICE EXTCODESIZE EXTCODECOPY RETURNDATASIZE RETURNDATACOPY EXTCODEHASH	Environment details
0x40, 0x41,...,0x45, 0x46	BLOCKHASH COINBASE TIMESTAMP NUMBER DIFFICULTY GASLIMIT CHAINID	Get block detail
0x47...0x4f	unused	unused
0x48	BASEFEE	Current block base fee
0x50, 0x51,...,0x5a, 0x5b	POP MLOAD MSTORE MSTORE8 SLOAD SSTORE JUMP JUMPI GETPC MSIZE GAS JUMPDEST	Stack, memory, storage, program counter, and flow operation
0x5c...0x5f	unused	unused
0x60, 0x61,...,0x7e, 0x7f	PUSH1 PUSH2 ... PUSH31 PUSH32	Push operation
0x80, 0x81,...,0x8e, 0x8f	DUP1 DUP2...DUP15 DUP16	Duplicate operation
0x90, 0x91,...,0x9e, 0x9f	SWAP1 SWAP2 ... SWAP15 SWAP16	Swap operation
0xa0, 0xa1,..., 0xa4	LOG0 ... LOG4	Log record operation
0xa5..0xaf	unused	unused
0xf0, 0xf1,...,0xf5	CREATE CALL CALLCODE RETURN DELEGATECALL CREATE2	System and call operation
0xf6...0xf9	unused	unused
0xfa	STATICCALL	Low level call
0xfb	unused	unused
0xfc, 0xfd,...,0xfe, 0xff	TXEXECGAS REVERT INVALID SELF-DESTRUCT	System, stop, halt operation

an attacker can re-enter the function after an external call is made to the contract, which happens when the internal state is not updated [148]. The *withdraw* function is seen as potentially risky in Figure 6.4 because it transfers Ether to the caller before

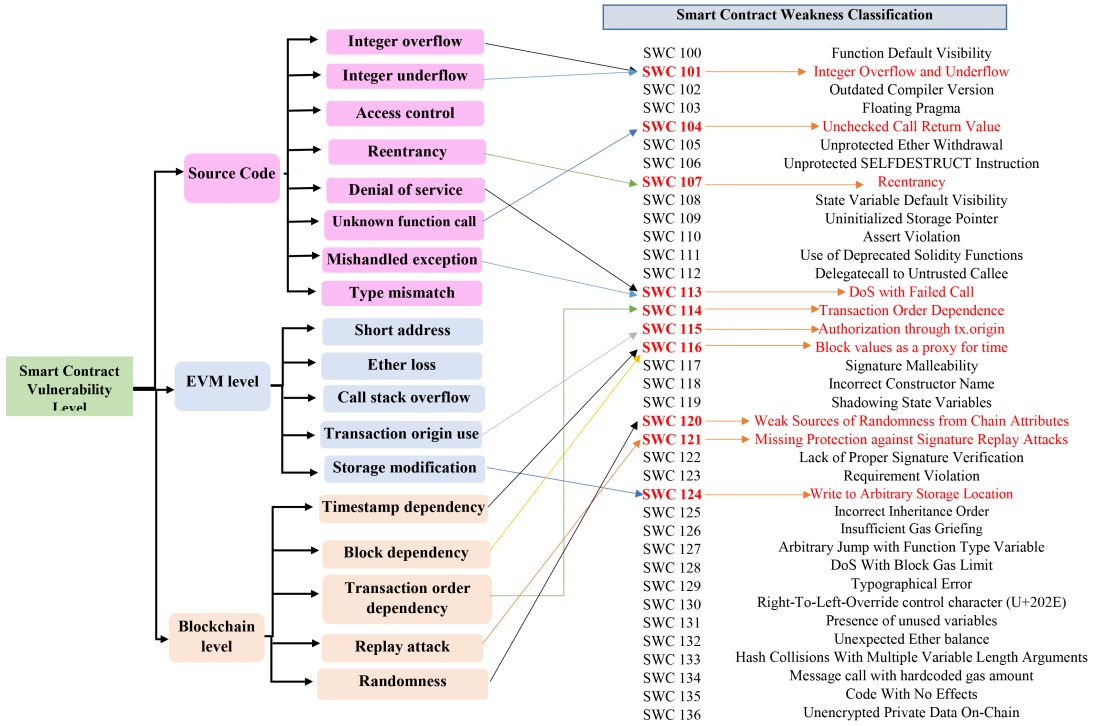


Figure 6.3: Categorization of smart contract vulnerabilities

```
contract Victim {
    mapping(address => uint) public balances;
    function deposit() public payable {
        balances[msg.sender] += msg.value;
    }
    function withdraw() public {
        uint amount = balances[msg.sender];
        require(amount > 0,"Insufficient balance");
        (bool success, ) = msg.sender.call{value: amount}("");
        require(success, "Transfer failed");
        balances[msg.sender] = 0;
    }
}
```

(a) Victim Contract

```
contract Attacker {
    Victim public victim;
    uint public count = 0;
    constructor(address _victim) {
        victim = Victim(_victim);
    }
    receive() external payable {
        if (count < 10) {
            count++;
            victim.withdraw();
        }
    }
    function attack() public payable {
        require(msg.value >= 1 ether, "Send at least 1 Ether");
        victim.deposit{value: msg.value}();
        victim.withdraw();
    }
}
```

(b) Attacker Contract

Figure 6.4: Example of Solidity contract to demonstrate a reentrancy attack

the balance is updated. An attacker can re-enter the *withdraw* function before the balance is set to zero. Attackers can deplete cash from the Victim contract by utilizing the *withdraw* method via the fallback function up to ten times before updating the balance.

[2] ***Integer Underflow/Overflow :*** Problems with underflow and overflow occur in Solidity and many other programming languages when mathematical operations go beyond the boundaries of the data type being utilized [138], [35]. Significant consequences, such as denial of service or illegal access to money, might emerge from such vulnerabilities.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;
contract VulnerableToken {
    mapping(address => uint256) public balances;
    function deposit(uint256 amount) public {
        balances[msg.sender] += amount;
    }
    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount, "Insufficient balance");
        balances[msg.sender] -= amount;
    }
}
```

Figure 6.5: Example of Solidity contract to demonstrate integer overflow/underflow vulnerability

Adding a particularly large amount of tokens (up to a maximum value of uint256) and a high current balance could lead to an overflow in the *deposit* function in Figure 6.5.

[3] ***Access Control :*** Unclear or inadequate checks on the necessary authorizations to carry out certain operations are the main cause of access control vulnerabilities in smart contracts. Without adequate enforcement of these permissions, unauthorized parties may take advantage of the contract by accessing variables or functions reserved for authorized users only. Illegal activities, such as money transfers, might occur due to the absence of supervision.

```
}

function initContract() public {
    owner = msg.sender;
}
```

Figure 6.6: Example of Solidity contract to demonstrate access control vulnerability

In Figure 6.6 (`https://dasp.co/#item-2`), if the contract permits any individual to invoke *initContract*, it may result in unauthorized modifications to the ownership. This function should typically be limited to execution only during the contract's deployment or by an authorized address.

[4] ***Unchecked call :*** Smart contracts are susceptible to serious vulnerabilities that can be introduced by unchecked calls in Solidity, including delegatecall(), call(), and send(). Instead of immediately rolling back failed transactions, these functions return

a boolean value indicating whether the operation is successful. Developers risk major security gaps if they don't verify these return values.

```
function withdraw(uint256 _amount) public {
    require(balances[msg.sender] >= _amount);
    balances[msg.sender] -= _amount;
    etherLeft -= _amount;
    msg.sender.send(_amount);
}
```

Figure 6.7: Example of Solidity contract to demonstrate unchecked low-level call vulnerability

In Figure 6.7, *withdraw* function contains a potential vulnerability due to the use of the *send*()method without checking its return value.

6.5 Proposed Methodology

In this section 6.5, we lay out the framework for our study, which includes a methodical strategy for pre-processing data, choosing models, and assessing their performance. The dataset comes from Hugging Face, which is generated by Rossini et al. (2022) and is shown in Figure 6.8. The Slither tool is used to analyze the corresponding bytecode and identify vulnerabilities. Then, as explained in Figure 6.12, the bytecode is converted into simplified opcodes.

After this adjustment, the feature matrix or input vector is then generated using *TfidfVectorizer*. We use oversampling techniques to balance the training set because the collected dataset has an uneven distribution of labels. Finally, we perform multi-label classification by training a model for each class separately. Access control is labeled as L0, arithmetic vulnerabilities as L1, other vulnerabilities as L2, reentrancy as L3, unchecked call vulnerabilities as L5, and safe contracts as L4. To find out if an instance is vulnerable or not, the model is tested against each label. The detailed process of our methodology is given below.

6.5.1 Data Pre-processing

Prior to conducting any analysis, it is essential to convert smart contracts into a format that facilitates feature extraction. The pre-processing step is vital as it guarantees that the data input into the detection algorithms is clean, relevant, and properly structured.

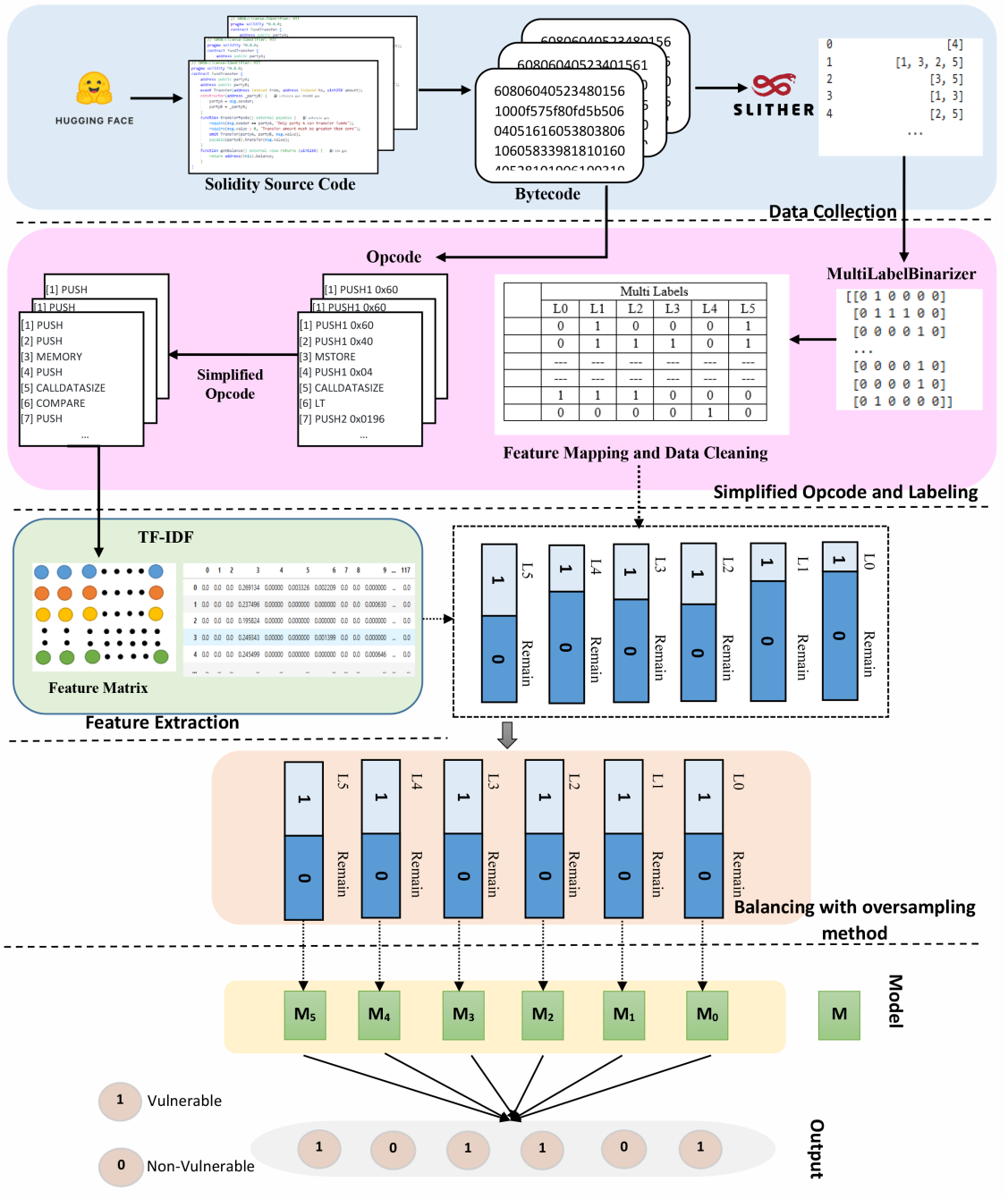


Figure 6.8: Workflow of proposed methodology

6.5.1.1 Dataset, Label

The dataset utilized in this study is sourced from Hugging Face [145] and comprises 106,474 smart contracts that have been audited using the Slither tool [149]. The dataset is partitioned into training, testing, and validation subsets, following an 8:1:1 distribution ratio. The training set comprises 79,641 smart contracts, the testing set consists of 15,972 smart contracts, and the validation set contains 10,861 smart contracts. Following an analysis of the label distribution within the dataset, SMOTE is implemented to address the class

imbalance issue. It enabled the creation of a new training set with a balanced representation of the labels, along with an updated testing set split, which is discussed in detail in subsubsection 6.5.1.4. The analysis presented in Figure 6.9, Figure 6.10 reveals that

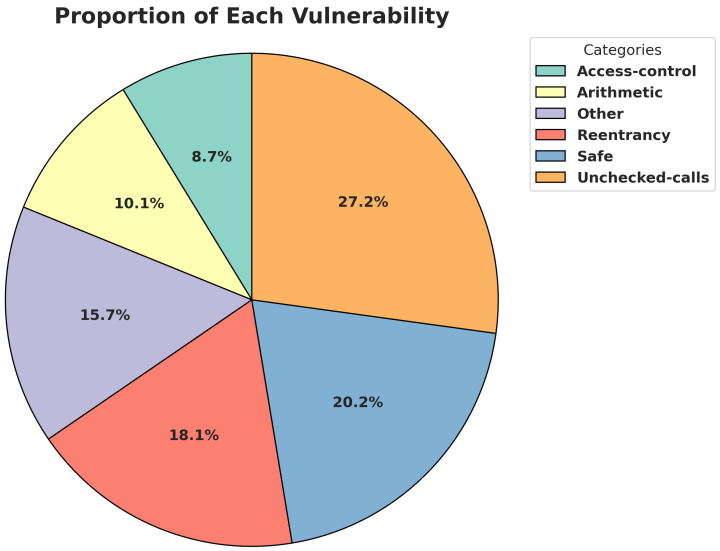


Figure 6.9: Distribution of categories in the dataset

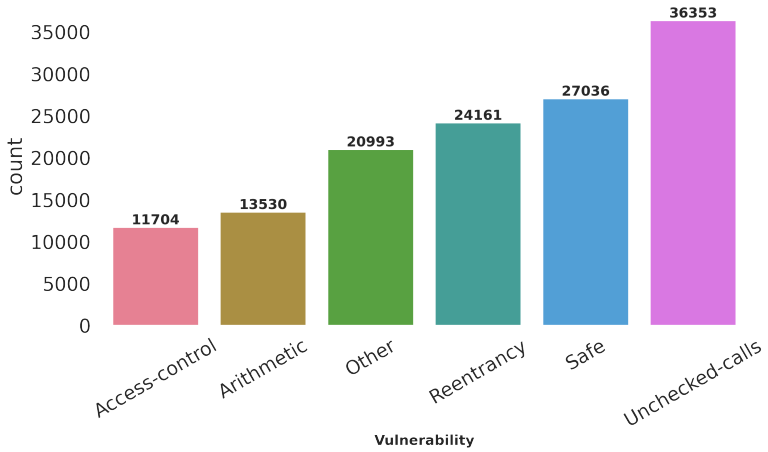


Figure 6.10: Number of samples for each category

five of the six categories identified are vulnerabilities, specifically access control, arithmetic, other, reentrancy, and unchecked call, with one category classified as safe. As detailed in subsection 6.4.2, a *access control* vulnerability arises when a smart contract inadequately limits access to its functions, permitting unauthorized users to perform essential operations [150]. Arithmetic vulnerabilities in smart contracts stem from integer underflow and overflow errors, posing significant risks when using unsigned integers. The category labeled as *Other* includes a range of Solidity detection issues, such as the identification of uninitialized state variables that can result in unforeseen behavior, along with erroneous equality

checks that could enable attackers to exploit conditions related to adequate Ether or token balances. [151]. Reentrancy happens when an external call to another contract allows it to re-enter the calling contract prior to the completion of the initial execution, which can result in unforeseen behavior [144]. Low-level calls, including call, delegatecall, and staticcall, do not automatically revert upon failure. They return a boolean value that signifies the success of the operation [151], [35], [150].

The collected dataset has 4 data fields, namely *address*, *source code*, *bytecode*, and *slither*.

[1] **Address** of the smart contract on Ethereum.

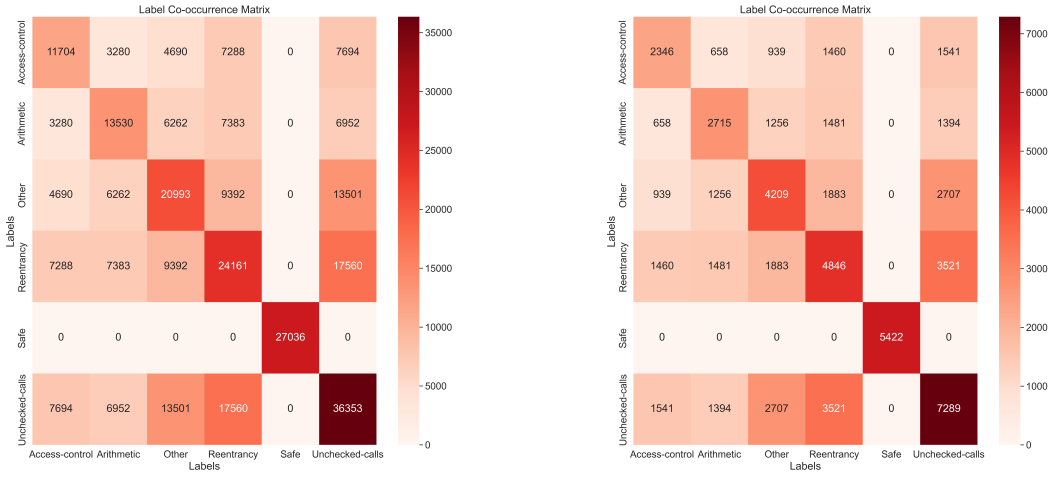
[2] **Source code** is the code of a smart contract written in Solidity.

[3] **Bytecode** is a string that represents the bytecode of the smart contract, which can be retrieved using web3.eth.getCode(). The string may be '0x' when this information is unavailable.

[4] **Slither** provides class labels. Each source code is analyzed using this static tool to get the class labels.

No additional work is required on the dataset's labels, as the dataset comes with predefined labels for each vulnerability.

The distribution of samples across each category in the training and test sets is depicted in Figure 6.11.



(a) Train Co-occurrence Matrix (b) Test Co-occurrence Matrix

Figure 6.11: Co-occurrence matrix for dataset

Contracts classified as "safe" exhibit no overlap with other categories. The training set exhibits a varied distribution of contract counts across different vulnerability classes, with access control being the least represented category. The classes exhibit considerable imbalance, with *reentrancy*, *safe*, and *unchecked calls* representing the largest sample sizes. The SMOTE algorithm is employed to mitigate the class imbalance in the training dataset. SMOTE produces synthetic samples for minority classes, aiding the balance of sample distribution across all categories.

This study employs multi-label binarization to convert each target label into a binary vector, enabling the model to handle multiple labels concurrently. The process is enabled by *MultiLabelBinarizer()* from scikit-learn [152], which is intended to identify distinct labels and transform them into a binary format. This representation indicates that each row corresponds to a sample, and each column denotes a specific type of vulnerability. This method provides a systematic framework for addressing multi-label classification tasks.

6.5.1.2 Opcode Simplification

As previously established in subsection 6.4.1, each instruction in the bytecode corresponds to an opcode, a fundamental command comprehensible by the EVM. Opcode offers a more direct, lower-level perspective on contract execution, enhancing the precision of vulnerability discovery relative to the analysis of bytecode alone. We transform the bytecode of the Ethereum smart contract into a more comprehensible format by disassembling it into opcodes, which may subsequently be classified into established categories. This classification streamlines the analytical process, facilitating the rapid detection of patterns and potential weaknesses, like reentrancy attacks or arithmetic overflows. Opcode simplification commences with the bytecode_to_opcode function (as illustrated in Figure 6.12)., which ac-

```
opcode_categories = {
    'ARITH': ['ADD', 'MUL', 'SUB', 'DIV', 'SDIV', 'SMOD', 'MOD', 'ADDMOD', 'MULMOD', 'EXP', 'SIGNEXTEND'],
    'COMPARE': ['LT', 'GT', 'SLT', 'SGT'],
    'LOGIC': ['AND', 'OR', 'XOR', 'NOT'],
    'MEMORY': ['MLOAD', 'MSTORE', 'SLOAD', 'SSTORE', 'MSIZE'],
    'RETURN': ['RETURN', 'REVERT', 'RETURNDATASIZE', 'RETURNDATACOPY'],
    'PUSH': ['PUSH1', 'PUSH2', 'PUSH3', 'PUSH4', 'PUSH5', 'PUSH6', 'PUSH7', 'PUSH8', 'PUSH9', 'PUSH10',
             'PUSH11', 'PUSH12', 'PUSH13', 'PUSH14', 'PUSH15', 'PUSH16', 'PUSH17', 'PUSH18', 'PUSH19', 'PUSH20',
             'PUSH21', 'PUSH22', 'PUSH23', 'PUSH24', 'PUSH25', 'PUSH26', 'PUSH27', 'PUSH28', 'PUSH29', 'PUSH30', 'PUSH31', 'PUSH32'],
    'DUP': ['DUP1', 'DUP2', 'DUP3', 'DUP4', 'DUP5', 'DUP6', 'DUP7', 'DUP8', 'DUP9', 'DUP10',
            'DUP11', 'DUP12', 'DUP13', 'DUP14', 'DUP15', 'DUP16'],
    'SWAP': ['SWAP1', 'SWAP2', 'SWAP3', 'SWAP4', 'SWAP5', 'SWAP6', 'SWAP7', 'SWAP8', 'SWAP9', 'SWAP10',
             'SWAP11', 'SWAP12', 'SWAP13', 'SWAP14', 'SWAP15', 'SWAP16'],
    'LOG': ['LOG0', 'LOG1', 'LOG2', 'LOG3', 'LOG4']
}
```

Figure 6.12: Opcode simplification

cepts a hexadecimal string denoting the bytecode, eliminates the '0x' prefix if applicable, and disassembles it into opcodes utilizing the disassemble_hex function from the *pyevmasm*

library. Subsequently, the simplify_opcode function classifies the disassembled opcodes into overarching categories, including arithmetic operations, comparisons, and memory operations, by verifying whether each opcode commences with any of the designated names in the opcode_categories dictionary.

6.5.1.3 Feature Extraction

The TF-IDF method is utilized to extract the feature vector from the opcode for model training. TF-IDF, or Term Frequency-Inverse Document Frequency, is a statistical metric employed to evaluate the importance of a word in a particular document compared to a broader set of documents, referred to as a corpus. Term frequency measures the frequency of a word's occurrence within a document, computed as:

$$\text{TF}(t, d) = \frac{\text{Count of term } t \text{ in document } d}{\text{Total count of terms in document } d} \tag{6.5.1}$$

In contrast, inverse document frequency evaluates the importance of a word across the entire corpus, calculated as:

$$\text{IDF}(t, D) = \log\left(\frac{\text{Total number of documents in the corpus } D}{\text{Number of documents that include term } t}\right) \tag{6.5.2}$$

By combining these two factors, the TF-IDF value for a term (t) in a document (d) is given by:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t, D) \tag{6.5.3}$$

Our method utilizes the *TfidfVectorizer* from scikit-learn [152] library to convert a set of text documents into a matrix of TF-IDF features. The vectorizer is set up with multiple parameters. The parameter decode_error='ignore' allows for the omission of decoding errors, lowercase=True standardizes text to lowercase, and min_df=2 restricts consideration to terms present in a minimum of two documents, effectively eliminating infrequent words. The fit method is utilized on the simplified_opcode to prepare the vectorizer by acquiring the vocabulary and IDF values from the text data. Transform method is then applied to convert the training and test data into TF-IDF feature matrices, yielding *X_data_transformed* for the training set and *X_test* for the test set. This transformation facilitates the application of machine learning algorithms to text data by converting it into a numerical format that reflects the significance of each term within the documents.

6.5.1.4 Data Balancing

The training and testing datasets, as outlined in subsubsection 6.5.1.1 and Figure 6.4, demonstrate a notable class imbalance that hinders the effective detection of vulnerabilities. The distribution of instances among various types of vulnerabilities, along with the "safe" category, exhibits significant variation, as demonstrated in Figure 6.13 and Figure 6.14. In order to resolve this issue, the training and testing datasets have been merged to enable the application of SMOTE on the consolidated dataset. The comprehensive and cohesive data set is partitioned into training and testing subsets.

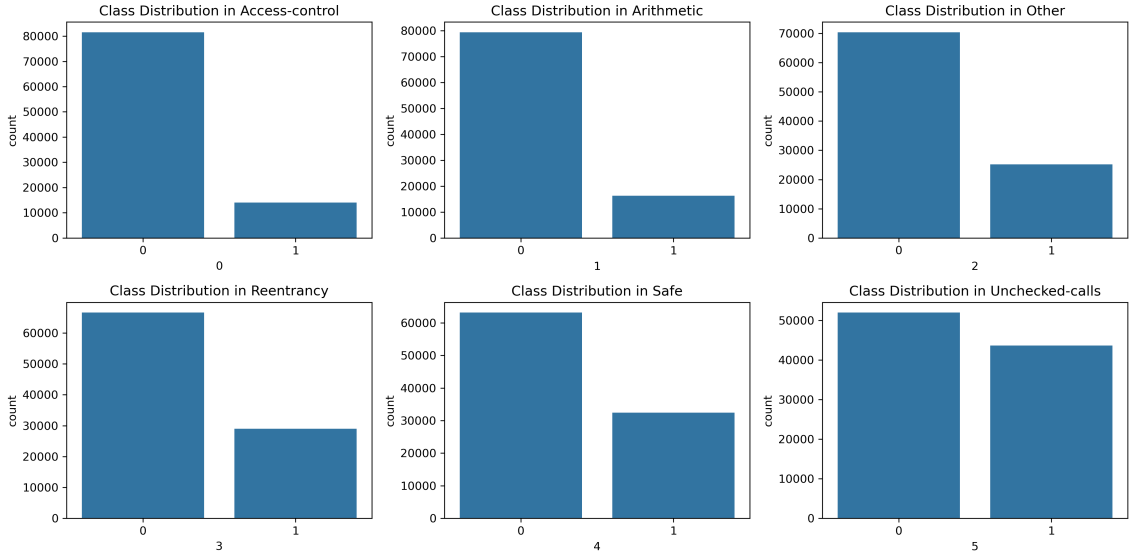


Figure 6.13: Distribution of each category

SMOTE has been utilized to enhance the representation of minority classes, thereby addressing the issue of class imbalance (as shown in Figure 6.15). SMOTE generates new examples by interpolating between closely situated minority class samples in the feature space rather than simply duplicating existing samples.

6.5.2 Training Models

We present SmarConTest, a vulnerability detection method based on stacked generalization. To demonstrate the efficacy of our method, we compare it to several state-of-the-art models:

[1] **LightGBM** is scalable and efficient, making it ideal for huge datasets. The system builds sequential decision trees to fix errors from previous ones [153]. LightGBM uses "leaf-wise" growth to focus on the biggest faults, improving accuracy and convergence

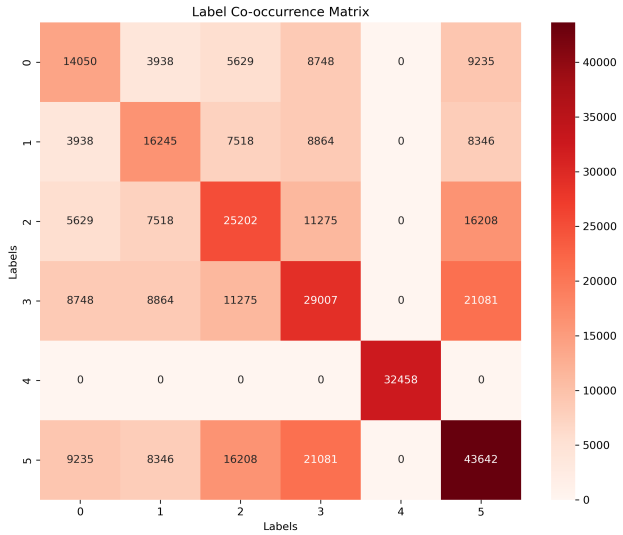


Figure 6.14: Co-occurrence matrix for unified dataset

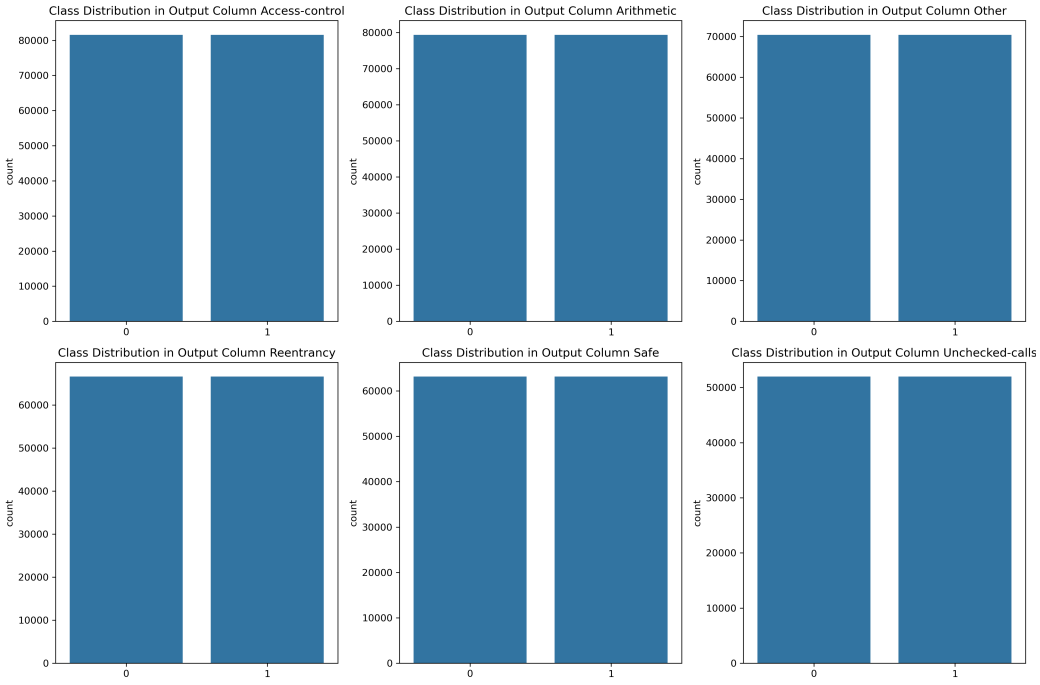


Figure 6.15: Balanced data samples

over level-wise techniques. Our system uses LightGBM to classify access control, Arithmetic, Other, Reentrancy, Safe, and unchecked calls. The model is trained with parameters such as boosting type, learning rate, and num_leaves for evaluation.

[2] **RF** trains multiple decision trees and combines their outputs to increase forecast accuracy and reduce overfitting. Each tree in the forest is trained on a random subset of data using bootstrap sampling, and at each split, only random characteristics are

examined. This randomization makes the model more resilient and less prone to overfit than a single decision tree.

[3] **DT** depict decisions and outcomes by recursively partitioning data based on feature values. Our model iterates over many labels, partitioning the dataset into training and testing sets for each. A DecisionTreeClassifier is trained on training data and then makes predictions for the test set.

[4] **KNN** classifies data points in the feature space based on the predominant class among their 'k' nearest neighbors. The model iteratively processes multiple labels, dividing the dataset into training and testing sets for each label. A *KNeighborsClassifier* is initialized with five neighbors and subsequently trained using the training data.

[5] **SVM** is employed for classification tasks and demonstrate notable efficacy in high-dimensional spaces. The fundamental concept of SVM involves identifying the optimal hyperplane that distinguishes various feature space classes while maximizing the margin between these classes. It is especially beneficial for datasets that lack linear separability. In this implementation, SVM is utilized to classify a set of resampled training examples into distinct categories.

[6] **XGBoost** is a boosting algorithm designed to improve both speed and model accuracy. The process is accomplished by reducing the differences between predicted values and actual outcomes, referred to as residual errors, and utilizing a regularized loss function to alleviate the risk of overfitting.

6.5.3 Performance Metrics

The performance of our proposed method is evaluated with primary metrics such as F1, precision, recall, and accuracy. In our study, we also assess performance metrics based on a 2-D confusion matrix, which is composed of four essential elements:FN,FP,TN,andTP.

The equations to compute performance measures are as below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.5.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.5.5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.5.6}$$

$$F1 \quad score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.5.7}$$

$$L_{hamming}(d, \widehat{d}) = \frac{1}{S_n L} \sum_{t=0}^{S_n - 1} \sum_{u=0}^{L-1} 1(d_{t,u} \neq \widehat{d}_{t,u}) \tag{6.5.8}$$

$$J(tl, pl = \frac{|tl \bigcap pl|)}{|tl \bigcup pl|)} \tag{6.5.9}$$

6.6 Experiment and Evaluation

This section 6.6 provides a comprehensive examination of the experiments carried out and an assessment of the results. The execution of SmarConTest is performed utilizing Python, with all experiments taking place in the operating environment outlined in Table 6.3. We aim to systematically explore various research questions related to SmarConTest by analyzing the experimental results.

Table 6.3: Experimental setup

Parameter	**Configuration**
Operating system	Windows 11 Pro
CPU	12th Gen Intel(R) Core(TM) i7-12700, 2.10 GHz
RAM	16.0 GB
TensorFlow	2.18.0
Keras	3.8.0
Scikit-learn	1.6.1

6.6.1 Research Questions

Our goal is to find answers to the following research questions:

[1] *RQ1 :* Does SmarConTest adequately detect the five vulnerabilities? How is its effectiveness?

[2] *RQ2 :* How does SmarConTest perform in comparison to other state-of-the-art ML techniques?

[3] *RQ3 :* How does SmarConTest compare to state-of-the-art ML and DL methods when tested on the same dataset?

[4] *RQ4 :* How efficient is SmarConTest in terms of time when it comes to identifying vulnerabilities?

6.6.2 Our Model

The stacking technique is utilized to pinpoint vulnerabilities in smart contracts. This method uses several base learners to enhance predictive accuracy. It employs the RF, DT, and XGBoost Classifiers as its foundational learners, as demonstrated in Figure 6.16. The RF classifier is an ensemble learning technique that effectively handles high-dimensional data while exhibiting robustness against overfitting. The Random Forest model has 100 estimators and a random state set to 42. The Decision Tree model employs a random state value of 42 (*random_state*) for its operations. The XGBoost Classifier is recognized for its scalability and efficiency, utilizing machine learning algorithms within the Gradient Boosting framework to achieve high performance and speed. The final estimator in the stacking model is a Logistic Regression classifier. This classifier is recognized for its interpretability and integrates the predictions from the base learners.

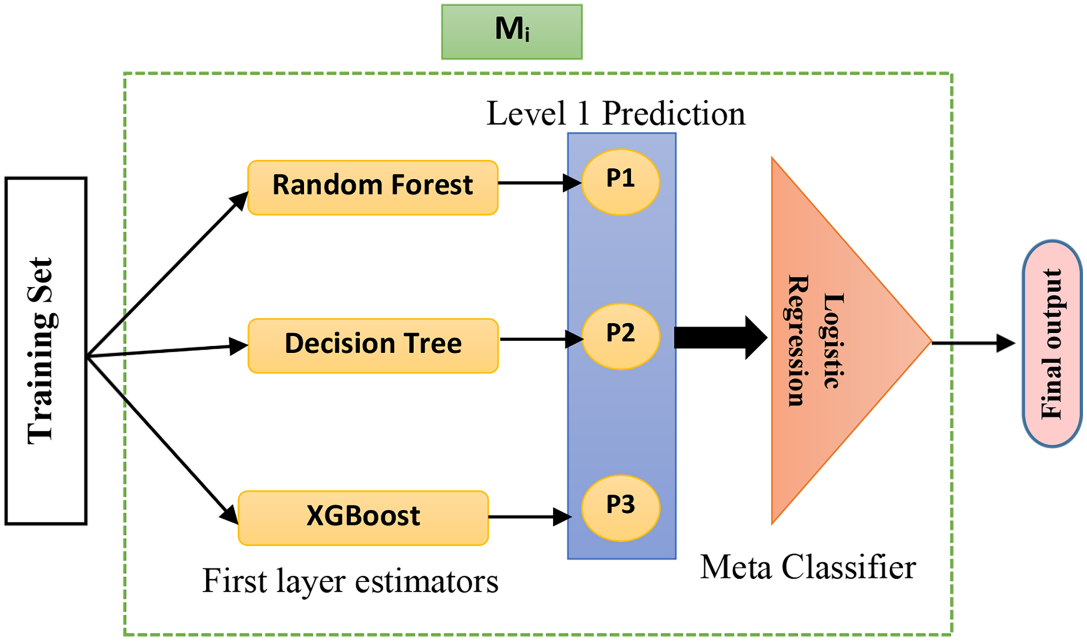


Figure 6.16: Model utilized for vulnerabilities detection

Data preparation involves the use of features categorized into six labels. To ensure that a substantial segment of the balanced dataset is allocated into training and testing sets at a 70-30 ratio, with a separate set designated for evaluation. The stacking classifier is trained on the training dataset, with each base learner fitted to the data and its predictions utilized

as input for the final estimator.

6.6.3 Performance Comparison and Discussion

To address *RQ1*, an experiment is conducted to detect five specific vulnerabilities and one non-vulnerability, as outlined in subsubsection 6.5.1.1. The models utilized for this detection task consist of LightGBM, RF, DT, KNN, XGBoost, and SVM. The methodology for training these models is detailed in section 6.5, which also includes a description of the use of TF-IDF for feature extraction.

Table 6.4: Performance comparison of various state-of-art methods for the detection of smart contract vulnerabilities

Type	**Performance**	**Methods**						
		LightGBM	RF	DT	KNN	SVM	XGBoost	**Stacking**
Access control	Accuracy (%)	90.56	97.28	93.02	93.36	71.20	94.36	**97.34**
	Precision(%)	90.57	97.28	93.04	93.76	71.58	94.36	**97.34**
	Recall (%)	89.99	97.30	94.04	**98.02**	78.06	93.77	97.45
	F1 (%)	90.56	97.28	93.02	93.34	71.04	94.36	**97.34**
	Hamming Loss	0.09	0.02	0.06	0.06	0.28	0.05	0.02
	Jaccard Similarity	0.82	0.94	0.86	0.87	0.55	0.89	0.94
Arithmetic	Accuracy (%)	87.92	**96.24**	90.80	92.70	72.22	92.89	**96.24**
	Precision(%)	87.92	**96.24**	90.81	93.26	72.30	92.89	**96.24**
	Recall(%)	88.32	96.41	91.64	**98.18**	69.38	92.29	96.11
	F1 (%)	87.92	**96.24**	90.79	92.67	72.20	92.89	**96.24**
	Hamming Loss	0.12	0.03	0.09	0.07	0.27	0.07	0.03
	Jaccard Similarity	0.78	0.92	0.83	0.86	0.56	0.86	0.92
Other	Accuracy(%)	86.00	93.56	88.29	89.43	66.03	90.16	**93.76**
	Precision(%)	86.09	93.59	88.30	89.97	66.07	90.16	**93.76**
	Recall (%)	88.48	94.82	89.15	**95.22**	68.76	90.33	94.14
	F1 (%)	85.99	93.56	88.29	89.39	66.00	90.16	**93.76**
	Hamming Loss	0.13	0.06	0.11	0.10	0.33	0.09	0.06
	Jaccard Similarity	0.75	0.87	0.79	0.80	0.49	0.82	0.88
Reentrancy	Accuracy (%)	89.48	94.60	90.40	91.16	77.88	92.54	**94.64**
	Precision(%)	89.48	94.60	90.41	91.37	77.90	92.54	**94.64**
	Recall(%)	89.02	94.02	91.21	**94.76**	79.37	91.84	94.40
	F1 (%)	89.48	94.60	90.40	91.15	77.87	92.54	**94.64**
	Hamming Loss	0.10	0.05	0.09	0.08	0.22	0.07	0.05
	Jaccard Similarity	0.80	0.89	0.82	0.83	0.63	0.86	0.89
Safe	Accuracy(%)	90.17	94.70	90.79	91.97	79.46	93.28	**94.75**
	Precision(%)	90.17	94.70	90.79	92.05	79.51	93.28	**94.75**
	Recall(%)	90.13	94.48	91.04	94.18	81.46	92.96	**94.59**
	F1 (%)	90.17	94.70	90.79	91.96	77.87	93.28	**94.75**
	Hamming Loss(%)	0.09	0.05	0.09	0.08	0.20	0.06	0.05
	Jaccard Similarity(%)	0.82	0.89	0.83	0.85	0.65	0.87	0.90
Unchecked-calls	Accuracy(%)	88.89	93.50	89.80	89.32	76.18	91.53	**93.55**
	Precision(%)	88.91	93.50	89.80	89.39	76.43	91.53	**93.55**
	Recall (%)	89.98	93.41	89.73	91.31	81.02	91.77	**93.61**
	F1(%)	88.89	93.50	89.80	89.32	76.12	91.53	**93.55**
	Hamming Loss	0.11	0.06	0.10	0.10	0.23	0.08	0.06
	Jaccard Similarity	0.80	0.87	0.81	0.80	0.61	0.84	0.87

To evaluate the performance of each model, we have compared several metrics, as presented

in Table 6.4. Among the models evaluated, RF, DT, and XGBoost exhibited enhanced performance across these metrics. As a result, these three models were chosen as the base learners for the stacking-based methodology. RF Classifier with 100 estimators, a DT, and an XGBoost Classifier configured with the *use_label_encoder* parameter set to false and the *eval_metric* parameter set to 'logloss'. The stacking model is developed utilizing a stacking classifier and employing logistic regression as the final estimator. It uses the *predict_proba* method for generating probability predictions and omits the original feature set from the stacking procedure.

The data illustrated in Table 6.4 demonstrates that this stacking method has enhanced the accuracy for each label while simultaneously decreasing the Hamming loss. The precision for each label is detailed below: access control achieves an effectiveness rate of 97.34%. Arithmetic vulnerability is recorded at 96.24%. Other vulnerability is noted at 93.76%. Reentrancy attacks reach an effectiveness of 96.64%. Unchecked call issues are at 93.55%. Non-vulnerable contracts demonstrate an accuracy of 94.75%.

> ***Response to RQ1*:** With an overall average accuracy of 95.05%, SmarConTest can identify the five listed vulnerabilities.

The answer to *RQ2* is evident from Figure 6.17, Figure 6.18, and Figure 6.19. The stacking technique exhibits enhanced accuracy in vulnerability detection when evaluated against other specified methods. The confusion matrices for six smart contract vulnerability classes are presented in Figure 6.17, utilizing a stacking ensemble model. The model exhibits robust overall performance, characterized by elevated true positive and true negative counts across all categories, alongside comparatively low instances of false positives and false negatives. The access-control and arithmetic classes demonstrate the highest accuracy levels, whereas the 'reentrancy' and 'other' categories exhibit a marginally increased rate of misclassifications. The results demonstrate the model's capability to accurately identify a diverse array of vulnerabilities present in smart contracts.

Additionally, Figure 6.19 presents the average metrics recorded for LightGBM, with an accuracy of 0.8884, an F1 score of 0.8883, precision at 0.8886, recall at 0.8932, and a hamming loss of 0.11. For RF, the accuracy is 0.9498, F1 score is 0.9498, precision is 0.9499, recall is 0.9507, and hamming loss is 0.05. In the case of DT, the accuracy is 0.9052, F1 score is 0.9052, precision is 0.9053, recall is 0.9113, and hamming loss is 0.09. For KNN, the accuracy is 0.9131, F1 score is 0.9131, precision is 0.9163, recall is 0.9528, and hamming

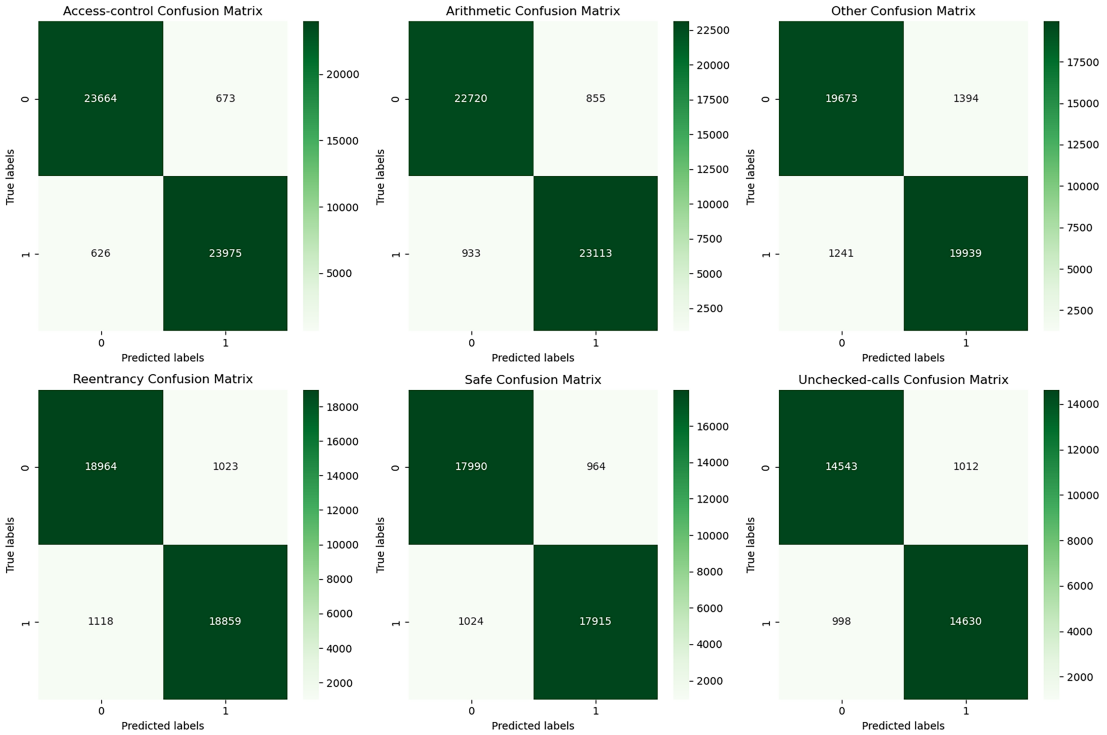


Figure 6.17: Confusion matrix

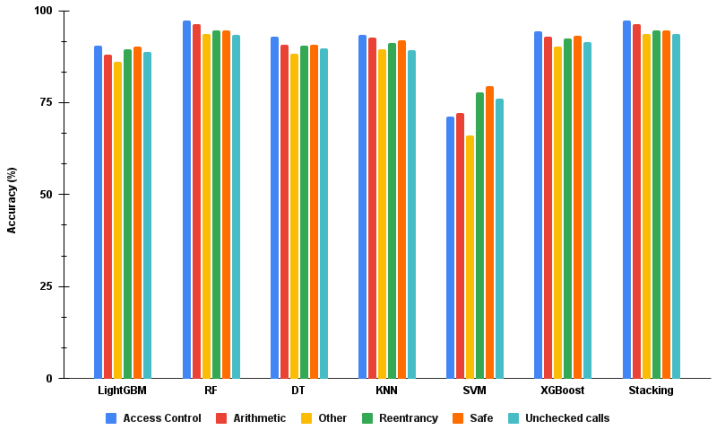


Figure 6.18: Comparison of various model accuracy across 6 labels

loss is 0.08. SVM shows an accuracy of 0.7383, F1 score of 0.7378, precision of 0.7397, recall of 0.7634, and a hamming loss of 0.26. XGBoost achieves an accuracy of 0.9246, F1 score of 0.9246, precision of 0.9246, recall of 0.9216, and a hamming loss of 0.07. Finally, the stacking technique yields an accuracy of 0.9505, F1 score of 0.9505, precision of 0.9501, recall of 0.9505, and a hamming loss of 0.04.

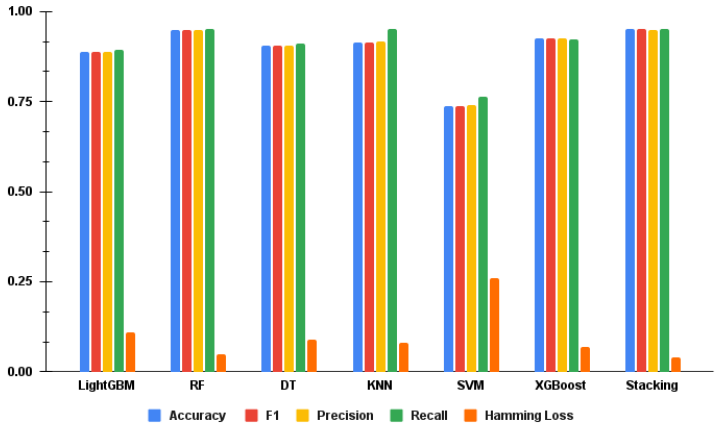


Figure 6.19: Cumulative comparison of models based on the average accuracy, precision, recall, F1 score, and hamming loss

> ***Response to RQ2*:** SmarConTest demonstrates better performance compared to all other models regarding hamming loss and accuracy, exhibiting the most significant advantage over SVM in accuracy.

In response to *RQ3*, we performed a benchmarking study employing state-of-the-art methods that utilize either ML or DL techniques, ensuring uniformity in evaluations using the same dataset. The performance of the proposed stacking technique for detecting reentrancy vulnerabilities is compared with the DistilBERT-based LSTM method described by Le et al. [154] in Figure 6.20 and Table 6.5, focusing on key metrics such as accuracy, recall, precision, and F1 score. It is important to note that direct experiments are not performed for this research question; rather, the values reported in the referenced studies are employed for comparison. The stacking model attained an accuracy of 94.64%, notably exceeding the 87.76% accuracy of the DistilBERT-based LSTM. The proposed technique achieves recall, precision, and F1 score values of 94.64%, 94.4%, and 94.64%, respectively. The DistilBERT-based LSTM method attained a recall of 88.38%, a precision of 86.06%, and an F1 score of 87.2%.

The accuracy results for several models in Figure 6.21 and Table 6.5 demonstrate a distinct performance hierarchy, with the suggested model achieving the best accuracy of 0.9505. Conversely, the ResNet1D model [155] attained an accuracy of 0.8381, whilst the ResNet and Inception models [155] achieved accuracies of 0.7928 and 0.8015, respectively. The LSTM baseline [155] and CNN model [156] demonstrated marginally worse performance, achieving accuracies of 0.7953 and 0.89, respectively. The RNN model [156] exhibited an

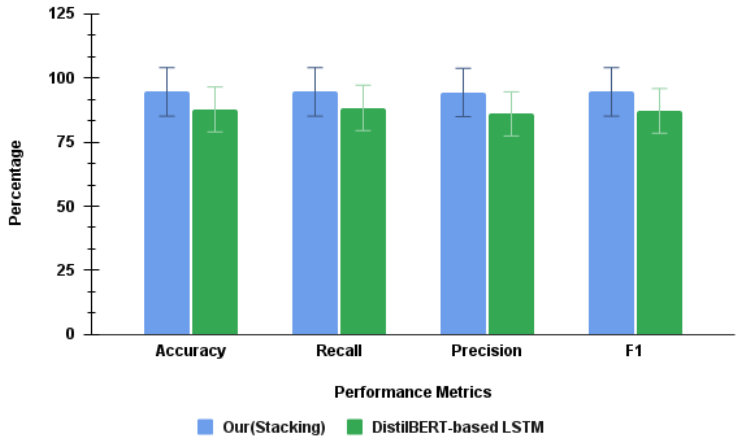


Figure 6.20: Performance comparison for reentrancy vulnerability

impressive accuracy of 0.93.

Table 6.5: Comparative analysis of our method compared to state-of-the-art approaches

Ref.	Data Set	Vulnerability/Type	Method	Accuracy	F1
[155]	Slither Audited Smart Contracts	Access control, arithmetic, miscellaneous, reentrancy, unchecked call, safe contract	ResNet1D ResNet Inception LSTM Baseline	73.53% 68.41% 69.88% 69.34%	83.81% 79.28% 80.15% 79.53%
[154]	Slither Audited Smart Contracts	Reentrancy	DistilBERT based LSTM	87.76%	87.2%
[156]	Slither Audited Smart Contracts	Access control, arithmetic, miscellaneous, reentrancy, unchecked call	RNN CNN	92.53% 88.63%	93% 89%
Our	Slither Audited Smart Contracts	Access control, arithmetic, miscellaneous, reentrancy, unchecked call, safe contract	Stacking	**95.05%**	**95.05%**

***Response to RQ3*:** SmarConTest demonstrated superior performance relative to state-of-the-art methods, achieving an accuracy of 94.64%, in contrast to the 87.76% accuracy of the DistilBERT-based LSTM. SmarConTest outperformed LSTM (0.7953), CNN (0.89), ResNet1D (0.8381), ResNet (0.7928), and Inception (0.8015) in the domain of vulnerability detection.

We determined the prediction time in order to answer *RQ4*. As mentioned in [150], for

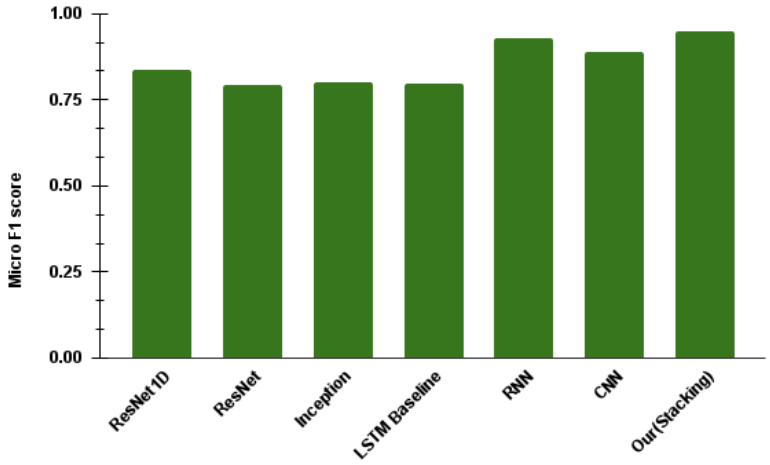


Figure 6.21: Performance comparison with state-of-art-approaches

each contract, Securify takes 2.8 seconds and Oyente 6.4 seconds to forecast vulnerability. The training period of our SmarConTest method is longer because it uses ensemble learning techniques. The model's efficiency in predicting many vulnerabilities within a single contract is outstanding once trained. As seen in Figure 6.22, it takes to anticipate these vulnerabilities is around 0.11 seconds.

```
[122]: mainApp()

+-----------------+------------+
|      Type       | Prediction |
+-----------------+------------+
| Access-control  |     No     |
|   Arithmetic    |     No     |
|      Other      |     No     |
|   Reentrancy    |     No     |
|      Safe       |    Yes     |
| Unchecked-calls |     No     |
+-----------------+------------+
Prediction time: 0.1152 seconds
```

Figure 6.22: Prediction time

***Response to RQ4*:** Upon providing the bytecode, our SmarConTest can detect multiple vulnerabilities in $\approx$ 0.11 seconds.

6.7 Summary

This paper presents SmarConTest, a methodology for detecting vulnerabilities in Ethereum smart contracts. SmarConTest employs a stacking method that amalgamates multiple base models to effectively identify various vulnerabilities, including access control vulnerabilities , arithmetic errors, reentrancy attacks, and unchecked calls, achieving an outstanding average accuracy of 95.05%. The methodology employed a balanced dataset enhanced by SMOTE, ensuring robust performance across several labels. Experimental findings demonstrated that SmarConTest outperforms existing state-of-the-art methods, showcasing improved accuracy and efficiency, with a prediction time of approximately 0.11 seconds per contract. The methodology showed significant advancements in vulnerability detection.

The next chapter introduces the ABHealChain framework, a blockchain-enabled healthcare system designed to ensure secure and scalable management of medical data. Built on HLF, ABHealChain employs ABAC to enforce fine-grained data access policies tailored to individual user roles and attributes. To address storage efficiency, the framework integrates IPFS for off-chain storage, ensuring that sensitive healthcare data is securely stored and efficiently retrieved, while maintaining the integrity and confidentiality of records through the blockchain.

Chapter 7

Blockchain-Enabled Healthcare System with Attribute-based Access Control

The purpose of this chapter * *is to highlight the numerous challenges in healthcare system. EHR security and privacy are important for healthcare professionals and patients. Healthcare system failures expose critical health data. The centralized database holds data, making it vulnerable to cyberattacks. We have used HLF with ABAC measures to ensure fine grained access and protection of medical data. Strong cryptographic protocols like AES-256 have reduced security concerns. Encrypted communications are sent across the network via this algorithm, restricting visibility to intended receivers and assuring data security. Our solution is carefully evaluated using Hyperledger Caliper. The evaluation covers critical performance parameters like latency, throughput, success rate, and resource utilization.*

7.1 Introduction

The decentralized architecture of the blockchain ledger guarantees that data processing is not restricted to a singular place, rendering it available and transparent to all participants within the network [3], [157]. Security of healthcare data necessitate explicit definitions of usage standards, especially in the realm of inter-agency data exchange [62].

The GDPR delineates fundamental standards for personal data handling, underscoring legality and transparency. A well-defined objective for data collecting is essential, accompanied by limitations on data processing beyond archiving purposes. Ensuring data accuracy

*The research work covered in this chapter has been accepted in: Anita Thakur, Virender Ranga and Ritu Agarwal, "ABHealChain: Enhancing Privacy and Security in Healthcare Data Sharing through Hyperledger Fabric and Attribute-based Access Control", —communicated
&
Anita Thakur, Virender Ranga and Ritu Agarwal, "Simulation-based Performance Evaluation of Consensus Algorithms in NS3 for Blockchain Network", *International Conference on Futuristic Technologies (INCOFT2025)*, Pune.

is vital, and prompt updates are necessary when required. The regulation requires that personal data be retained in a form that allows for identifying data subjects solely for the required length [158]. Integrating blockchain into healthcare systems protects data from unauthorized alterations and efficiently mitigates breaches. This technological advancement maintains data integrity, ensuring exceptional reliability [49]. Blockchain secures healthcare data by ensuring its permanence, enhancing security through cryptographic techniques, and decentralizing data storage across a network of nodes. Access to healthcare records can be rigorously controlled, enabling patients to confer or withdraw permissions as necessary [6]. The explicit audit trail of transactions and data alterations improves accountability and transparency.

Smart contracts enable consent management, whereas blockchain enhances interoperable data exchange and reduces fraud in insurance claims. Patients generally retain ownership of their health data, facilitating greater control over its use. Incorporating these features establishes blockchain as a robust mechanism for protecting sensitive healthcare data while preventing unauthorized access and manipulation [159]. A healthcare collaboration platform is a system intended to facilitate communication among various healthcare stakeholders, including physicians, regulatory bodies, healthcare providers, laboratories, patients, and pharmaceutical companies [160].

The evolving realm of the IoT and proliferation of health-related gadgets and mobile healthcare applications collect and transmit substantial medical data daily [161]. Effective data flow management is crucial for privacy and improving security protocols. Blockchain provides a solution by facilitating secure archiving and exchange of medical records while establishing a framework that confers ownership rights to people over their medical data, thereby ensuring privacy of their health information [162].

7.2 Problem Statement

The healthcare sector faces considerable challenges in safeguarding the privacy and security of shared medical data. Data breaches, unauthorized access, and compromised patient information undermine trust and lead to financial setbacks. Although blockchain technology holds promise for addressing these challenges, its application in healthcare faces scalability, security, and performance issues. Public blockchain frequently demonstrates significant delays, insufficient access control features, and slow transaction processing. This study seeks to present a solution that addresses these challenges and promotes the efficient incorporation

of permissioned blockchain in medical data.

7.3 Research Contributions

The research has yielded significant contributions, which encompass the following key aspects:

[1] Introduced a framework called ABHealChain that employs a permissioned blockchain to ensure the secure and transparent storage, and sharing of sensitive healthcare information among different healthcare entities.

[2] Utilizing ABAC and chaincode to protects against unauthorized alterations or distribution without the required consent and approval. ABAC operates by granting access based on the attributes supplied by the requesting entity. Access is granted when the requesting entity aligns with the defined access policy, ensuring the protection and privacy of the medical data.

[3] A symmetric key cryptography algorithm is employed to secure health data and provide robust protection. The encrypted information is then securely kept in off-chain storage. The entrusted recipient performs decryption while retrieving data, enabling access to the permitted information.

[4] A distributed off-chain storage system is employed to improve the scalability and security of data storage. Storing encrypted data on the blockchain protects sensitive information and alleviates blockchain network from managing substantial data volumes.

[5] The framework's effectiveness has been evaluated using the Hyperledger Caliper tool. The system's performance has been examined across different scenarios, emphasizing essential metrics like resource utilization, throughput, and latency.

7.4 Framework of proposed ABHealChain

This section provides an in-depth exploration of every element within the proposed framework. The entities shown in Figure 7.1 show the clients engaging with the framework. Within Hyperledger network, the chaincode is denoted as $CCName$. Each organizational entity has a certificate authority and peers, with the peers responsible for managing ledger instances and executing chaincode modules.

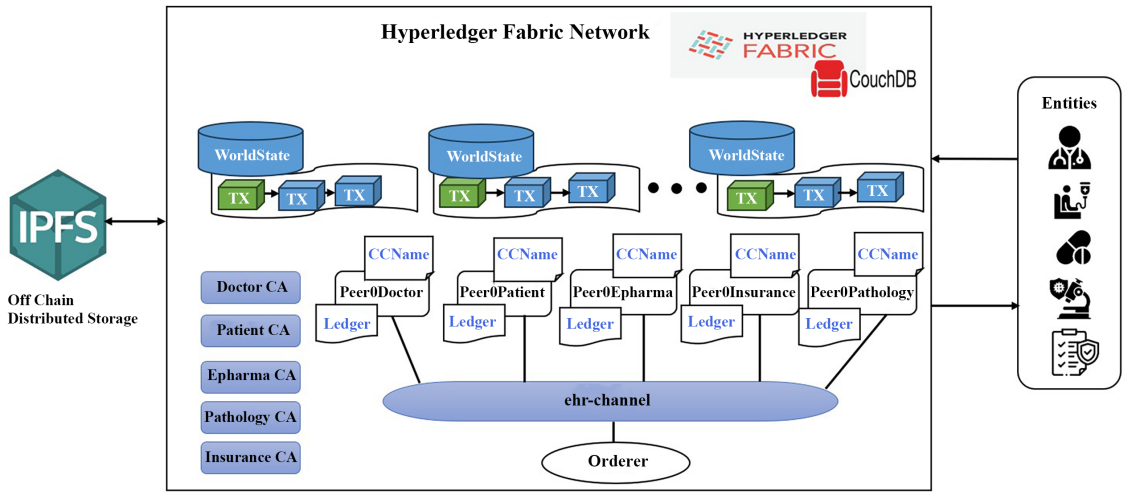


Figure 7.1: A broader illustration of the proposed framework

7.4.1 Components

The following discussion provides a high-level outline of the ABHealChain framework's foundational elements:

7.4.1.1 HLF

The Hyperledger blockchain network works within a controlled system, necessitating user registration for access. This network regulates authorization through Hyperledger modeling and an access control list. HLF functions as a fundamental structure for distributed ledger solutions, characterized by its modular design that facilitates the attainment of elevated levels of confidentiality, resilience, and flexibility [163], [133]. This investigation utilizes HLF as the selected permissioned blockchain framework, comprising designated entities, to facilitate the secure and decentralized sharing of healthcare data independent of a central authority. Given the delicate nature of medical information, which carries significant social and legal ramifications, adopting a closed blockchain system like HLF is essential for upholding the requisite privacy standards for this application. HLF offers a sophisticated method for overseeing access to healthcare records, enabling data owners to retain precise control over the disclosed data elements.

7.4.1.2 Consensus Mechanism

The consensus mechanism is an essential and foundational aspect of blockchain, a vital layer governing transaction validation. This mechanism employs the smart contracts layer to

authenticate and record transactions in the ledger, adhering to their sequential arrangement. A significant finding has emerged in our system, where differing values of the same attribute among network peers prevent the achievement of consensus. This consensus mechanism is essential for ensuring that all nodes in the network reach an agreement, thereby preserving the system's integrity and protecting the immutability of data.

7.4.2 Attribute-based access control

In HLF, ABAC limits access exclusively to users with the required attributes in their certificate. ABAC enables organizations to grant specialized access rights to specific users without necessitating their classification as network administrators. ABAC is an intricate access control model that determines access authorization by assessing multiple attributes, including specific permissions, environmental conditions, object properties, and requested user characteristics [164]. This model evaluates access requests by analyzing the presence or absence of essential attributes to decide whether to grant or deny access to the requester. Moreover, these traits can be leveraged by the designated entity to formulate particular access policies, aiding in assessing whether the requester possesses adequate access rights. ABAC is typically characterized by a set of five components, defined as follows: ABAC = $A_{ttr} \in \{S_j, O_j, O_p, E_c, D_p\}$ [165], [166].

Entity characteristics are represented by S_j during access requests. It helps identify and distinguish the entity requesting access. A patient's ID, name, address, contact information, and age may be included.

Resource characteristics, such as identity, ownership, confidentiality categorization, and application-specific qualities, are represented by O_j. The characteristics of an object—the features of the accessed resource—should also be detailed. These traits are usually assigned at formation.

O_p represents the user's requested action and its relationship to the resource. Read, update, and delete operations could be customized for specific resources. It establishes unambiguous user control tiers over the resource.

The E_c variable contains information about the environment at the moment of access request generation, including timestamps.

D_p represents the process of making a decision, particularly in assessing whether to approve or reject an incoming access request.

Within the framework of HLF, ABAC functions as the mechanism utilized to govern access permissions, thereby restricting entry solely to users who possess the necessary attributes embedded within their digital certificates.

7.4.3 InterPlanetary File System

A potential limitation exists in the storage capacity of blockchain infrastructure. Storing large volumes of data on the blockchain can result in scalability challenges. The diagnostic procedure encompasses multiple components, including images, medical reports, and laboratory findings, necessitating substantial storage capacity. To address this issue, off-chain storage can be utilized, integrating only the hash values of individual blocks into the blockchain to ensure data integrity. The primary data blocks will be maintained exclusively in off-chain storage. IPFS [167] has been employed as an off-chain repository for storing encrypted health-related information, as illustrated in Figure 7.2. IPFS does not function as a provider of storage solutions. This system is designed to enable decentralized data storage and sharing. Utilizing IPFS for data storage instead of blockchain is grounded in recognizing that the IPFS framework inherently manages substantial data volumes produced by multiple devices with greater efficiency. When data surpasses a specified threshold, typically greater than 256KB, IPFS allocates the encrypted information across various nodes. The storage of substantial data within the blockchain ecosystem may pose challenges related to system scalability.

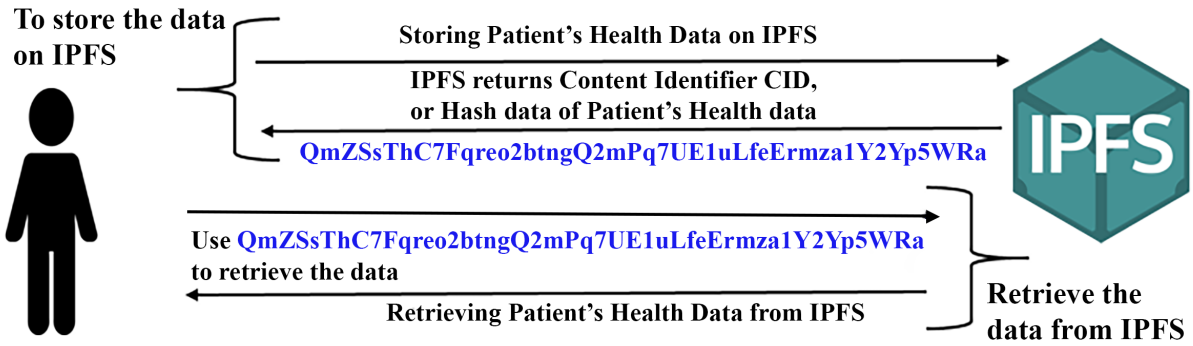


Figure 7.2: IPFS to safely store and retrieve medical records

7.4.4 Advanced Encryption Standard

AES is a widely adopted symmetric encryption algorithm established by the NIST in 2001 to provide secure data encryption. It operates on fixed-size blocks of data and supports key lengths of 128, 192, and 256 bits. Below is a overview of the AES encryption process:

Key Expansion: The AES algorithm begins by expanding the original encryption key into a series of round keys. The number of rounds varies based on the key size: 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key.

Initial Round:In the initial round of AES, the plaintext is XORed with the initial round key.

Rounds: AES processes the data through several rounds, each consisting of four essential steps:

[1] *SubBytes:* AES processes the data through several rounds, each consisting of four essential steps:

[2] *ShiftRows:* The rows of the state matrix are shifted left by varying numbers of bytes. The first row remains unchanged, the second row is shifted by one byte, the third by two bytes, and the fourth by three bytes.

[3] *MixColumns:* This step involves matrix multiplication, where each column of the state matrix is multiplied by a fixed matrix and reduced modulo a polynomial. This provides diffusion, enhancing the security of the encryption.

[4] *AddRoundKey:* The round key is XORed with the state matrix during this step. The round keys are derived from the original key during the key expansion phase.

Final Round: The final round is similar to the previous rounds but omits the MixColumns transformation. It consists of the SubBytes, ShiftRows, and AddRoundKey steps.

Result: After completing all rounds, the state matrix contains the encrypted data.

Decryption: The decryption process for AES is the reverse of the encryption process. The ciphertext is processed through the inverse transformations in reverse order, using the round keys in reverse order. The decryption steps include InverseSubBytes, InverseShiftRows, InverseMixColumns (except in the last round), and InverseAddRoundKey.

This encryption scheme process ensures both data confidentiality and security.

7.5 System Design

In this section, we have shown the ABHealChain framework components, emphasizing the desired security outcomes.

7.5.1 Security Goal/Threat Model

Blockchain provides a range of features to the framework in the healthcare industry, including privacy, security, integrity, access control, data exchange, decentralization, and mobility. The established system seeks to address the various issues in the area, with the following aims to be achieved.

[1] ***Privacy Preservation:*** Patient privacy presents the foremost challenge within this system due to the incorporation of vast personal and medical information. Irrespective of the organization requesting access to patients' health data, rigorous controls or restrictions on information are essential.

[2] ***Anonymity:*** Increasing patient data sharing for secondary uses, especially research, has gained popularity. This environment has struggled to establish data sharing due to patient privacy concerns. Except for legal exclusions, data custodians can share patient data for secondary purposes by gaining informed consent and anonymizing it. When medical data is collected, prior approval for unexpected studies is often impractical. In contrast, anonymization involves removing identifying elements and generalizing any residual data that may be used to identify the individual.

[3] ***Security:*** Data users are required to engage with the access control mechanism when obtaining patients' health data. The searching and querying procedures must concurrently ensure the absence of information leakage. In a healthcare blockchain ecosystem, users are responsible for verifying the authenticity of medical records during a transaction. The validation process involves confirming the authenticity of the record, ensuring it is sourced from a legitimate owner, and verifying its integrity to eliminate any potential for forgery.

[4] ***Access Control:*** Patients should limit the disclosure of their medical records and linked data, especially in the healthcare blockchain system. Medical data is only accessible to authorized persons; rigorous access controls are part of minimum disclosure. It also restricts authorized users to medical data relevant to their legal and vital

needs, reducing sensitive information exposure. Our research achieves these goals via ABAC techniques. Unauthorized access is prevented, boosting medical data security and privacy in the healthcare blockchain context.

[5] ***Tamper-proof:*** Tampering health data involves unauthorized or malicious changes, modifications, or falsifications of medical or healthcare records. It can significantly impact patients, healthcare providers, and the healthcare system. The system implementation involves defining constraints for creating, modifying, deleting, and retrieving patient records. Access to these operations is limited to authorized users with the necessary permissions.

[6] ***No Deliberate or Accidental Loss of Data:*** Unintentional exposure, deletion, or mishandling of patient data occurs in healthcare. Human error, system failure, data breach, inadequate backups, insider threat, or malicious assault might cause it. We used on-chain and off-chain storage methods to protect patient records, assuring data availability and security.

[7] ***Data Integrity:*** External blockchain entities cannot interfere with transactions that include medical data. Hash-chained storage protects. Data integrity and dependability are needed to protect patient records from unauthorized changes.

7.5.2 Proposed Architectural framework

The proposed framework utilizes a permissioned blockchain, specifically HLF, and integrates ABAC, an encryption scheme, and off-chain storage. The primary objective is to improve network communication and facilitate data sharing among individuals through access control policies embedded as smart contracts. This methodology accelerates the data-sharing process and reduces the likelihood of human errors. HLF allows various stakeholders in the healthcare sector to request authorization to access and formally interact with medical records. Each transaction involving all parties is executed with high security and auditability and subsequently recorded as a verifiable entry on a decentralized ledger.

As shown in Figure 7.3, the client (submitting peer) initiates a transaction and transmits a proposal to the endorsing peers. The proposal outlines the specifics of the proposed transaction, including the chaincode to be executed and the input parameters involved. The chaincode comprises functions for creating, deleting, updating, and retrieving records related to patients, doctors, pharmacies, path labs, and patient insurance. The framework comprises

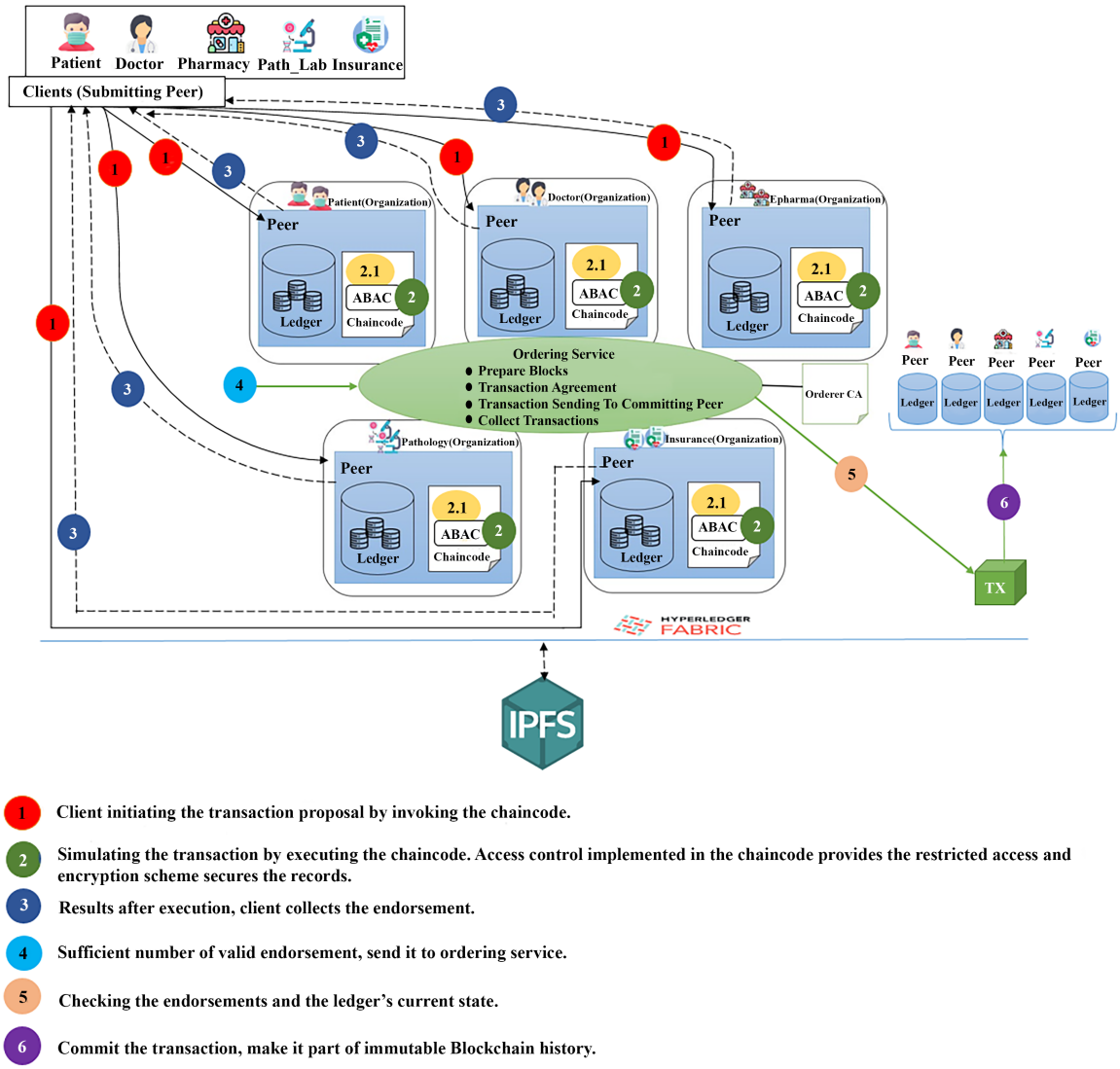


Figure 7.3: System architecture of ABHealChain: A blockchain-enabled healthcare system to manage medical data

five entities: *Patient*, *Doctor*, *Epharma*, *Pathology*, and *Insurance*. Members within the organization are tasked with simulating the transaction through the chaincode execution to ascertain its outcome. The peers have not updated the ledger with the transaction; they merely endorse its validity. ABAC is implemented in the system to address challenges associated with access control. Each user is granted exclusive access to the medical records of patients for whom they provide care.

ABAC enables organizations to create a detailed and precise framework for access control. Endorsing peers execute the basic chaincode $CCName$. If the transaction proposal satisfies the criteria of the endorsement policy, the peers reach a consensus on the transaction's validity. The client gathers endorsements from endorsing peers and submits them as part

of the transaction proposal response. Upon gathering an adequate number of valid endorsements according to the endorsement policy, the client compiles the endorsed transaction and submits it to the ordering service. The ordering service incorporates the transaction into a block and disseminates it to all peers within the network. Upon verification of all elements, the transaction is recorded in the ledger, thereby integrating it into the immutable blockchain history.

The implementation of IPFS in our system provides significant scalability advantages. This method differs from the direct storage of primary data or assets on the blockchain. The process begins with the encryption of the data, which is then transmitted to the IPFS network. IPFS subsequently generates the CID for the assets stored on the blockchain as hash data.

7.5.2.1 Participant in the System

Data users are individuals or entities seeking to obtain clinical or medical records pertaining to a patient's medical history. It includes information on prescribed medications, administered tests, generated or pending reports, and the status of insurance claims, whether approved or denied. The ABAC mechanism enables authorized users to access data according to their granted access rights. The entities involved in our system include $Patient$, $Doctor$, $Epharma$, $Pathology$, and $Insurance$.

7.5.2.2 Medical Record

Authorized healthcare participants who follow access control methods may be able to retrieve medical data, which includes information and assets belonging to the patient, doctor, and other relevant entities.

7.5.2.3 Chaincode

The chaincode deployed in the system provides participants with four fundamental functionalities: asset creation, asset deletion, asset retrieval, and asset modification. A participating entity, such as a physician, is provided access to critical assets, including $PatientID$, $DoctorID$, $DoctorName$, $Speciality$, and $DaysOfWorking$.

The $PatientID$ option is a unique identifier for the patient linked to the doctor. It is essential to emphasize that a physician must obtain authorization from the data owner, usually the patient, to access medical information, including history, diagnosis, and test results. Such permissions are essential for preserving data privacy and confidentiality within

the system.

7.5.2.4 Certificate Authority

In HLF, the CA is an essential element overseeing network participants' identities. CA is a reliable organization that provides digital certificates and cryptographic keys to network participants, such as users, peers, and orderers. The responsibilities of the CA in HLF encompass identity management, authentication, certificate revocation, renewal, and key pair generation.

7.5.2.5 Data Storage

The simultaneous pursuit of privacy and transparency poses significant challenges within a blockchain framework. Integrating data onto the blockchain presents notable issues regarding confidentiality and scalability. Off-chain storage systems will be responsible for storing comprehensive data. We adopted the IPFS protocol, a peer-to-peer distributed system intended to connect all computational devices within a fault-tolerant file system. This approach seeks to maintain privacy and ensure the reliability of data authenticity Figure 7.4.

Figure 7.4: Data stored on IPFS

7.6 Implementation

Numerous blockchain platforms have emerged, including significant examples such as Bitcoin, Ethereum, and Hyperledger. This study utilized HLF as the foundation for developing the ABHealChain system. The proposed HLF-based medical record solution ensures unalterable information for all relevant parties, eliminating the need for a centralized governing body within the healthcare domain. In this system, end-users, especially patients and other stakeholders, can access patient medical records, provided they have the necessary authorization via a Patient ID. Stakeholders must be authorized to read, delete, or update any data.

This framework is developed by integrating network entities and chaincode, employing the AES encryption algorithm for data protection Figure 7.5, implementing ABAC Figure 7.6, and utilizing IPFS.

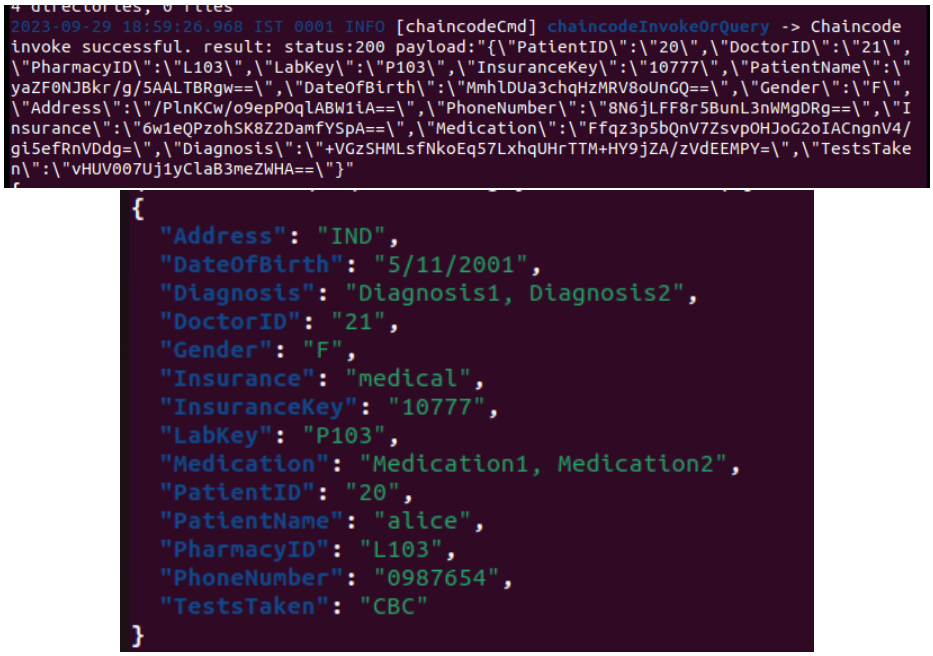


Figure 7.5: AES algorithm to encrypt the sensitive information

The framework is implemented on the HLF v2.5.0 system using an AMD RYZEN 5000 series processor, with IPFS serving as the off-chain database and CouchDB as the on-chain database. The chaincode is developed in JavaScript, and Hyperledger Caliper v0.6.0 is employed to assess the framework's performance.

This segment outlines the implementation of our proposed healthcare system utilizing HLF, designed to enhance the processing and execution of transactions among various stakeholders

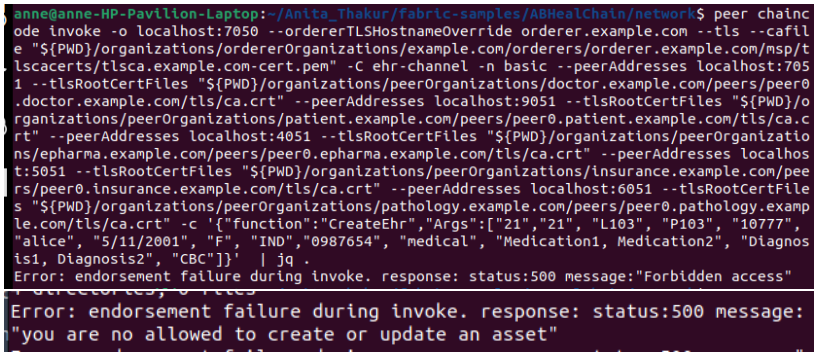


Figure 7.6: Restricting the access of unauthorized participants

within the blockchain network Table 7.1. It is achieved by implementing an execute-order-validate architecture within the Fabric framework. The Docker engine is employed in the configuration and launch of HLF. Docker is a containerization technology that operates at the level of the operating system, making it particularly advantageous for developers and system administrators. The fundamental efficacy is evident in its capacity to streamline the generation, deployment, and execution of applications or business networks intricately associated with the Hyperledger framework. Docker enables developers to consolidate all necessary dependencies and functionalities within a single containerized environment.

During the initial phase of ABHealChain, all stakeholders are required to complete the registration and enrolment process within the network. The system architecture is based on established practices, where the patient holds their medical records, the attending physician has access to relevant patient information, the insurance company retains certain data, the pharmacy manages medication details for the patient, and the pathology lab keeps records of the tests selected by the patient. Every participant in the network maintains authority over their records and provides access to other participants or users according to established privilege conditions. The implementation utilized the Client Identity "cid" library, enabling ABAC for chaincode. The "cid" library includes key functions like "GetId," "GetAttributeValue," and "AssertAttributeValue," facilitating efficient interaction with asset attributes.

The proposed system encompasses creating, updating, retrieving, and deleting records. When a medical record is initiated, it activates a transaction associated with the specified task in the chaincode, "CreateEhr." This transaction is then submitted to the network's peers. The proposal request for this transaction is routed to the endorsing party, as dictated by the endorsement policy established in HLF. The "CreateEhr" transaction includes parameters pertinent to the patient's medical record, such as P_{ID}, D_{ID}, Ph_{ID}, L_{key}, Ins_{Key}, P_{name}, DOB, $Gender$, $Addr$, $PhNo$, Ins, Med, $Diag$ and $Test_{Tn}$.

Table 7.1: Notation with their explanation

Notation	Definition
Ehr	Patient Record
DR_n	Doctor Record
PH_n	Pharmacy Record
LB_n	Lab Record
INS_n	Insurance Record
P_{ID}	Unique Identity of the patient/ PatientID
D_{ID}	Doctor ID
Ph_{ID}	Pharmacy ID of the associated patient
L_{key}	LabKey of associated patient/ Lab ID
Ins_{Key}	InsuranceKey of associate patient/ Insurance ID
P_{name}	Patient Name
D_{name}	Doctor Name
DOB	DateOfBirth
$Gender$	Gender
$Addr$	Address
$PhNo$	Phone Number
Ins	Insurance
Med	Medication
$Diag$	Diagnosis
$Test_{Tn}$	Tests Taken
D_{Spec}	Doctor Specialization
D_{Work}	Doctor Days of Working
C_{at}	Timestamp at pharmacy report of patient created
U_{at}	Updated At
L_{test}	Lab Tests
T_{status}	Status of Lab test
Doc_{ID}	Insurance DocumentID of Patient
Adh_{No}	Adhaar Number of Patient
$Amount$	Insurance Amount patient claimed
$Status$	Status if the amount claimed accepted or rejected

This transaction proposal is subsequently verified and validated according to the endorsement policy by the available and enrolled peer nodes within the Fabric blockchain network. The subsequent actions adhere to the execution sequence outlined in section 4. This healthcare solution provides a flexible platform for the initiation and deployment of chaincodes, which are crucial for the execution of various functions and procedures within the system.

The functionalities include creating, deleting, and retrieving essential healthcare records, such as patients, doctors, laboratories, pharmacies, and insurance. The system facilitates the efficient updating of records, ensuring the availability of the most current and accurate information.

The HLF blockchain network implements a stringent procedure requiring the majority of peer nodes to verify and validate each transaction proposal. Information blocks may be stored in the ledger following verification and validation processes. The efficacy of blockchain is rooted in its capacity to uphold reliable, secure, and trustworthy records. The trustworthiness of these records is established through rigorous verification, validation, and security measures. This technology enables trust among stakeholders through record storage, consensus mechanisms, private key utilization, and the formation of decentralized networks. These features facilitate secure and transparent communication among untrusted parties. This document provides a straightforward overview of our chaincode function, which is responsible for data storage in the ledger.

Algorithm 7.1 Pseudocode of CreateEhr function

Require: P_{ID}, D_{ID}, Ph_{ID}, L_{key}, Ins_{Key}, P_{name}, DOB, $Gender$, $Addr$, $PhNo$, Ins, Med, $Diag$, $Test_{Tn}$
Ensure: Patient Record Ehr Created
1: **Steps:**
2: **if** P_{ID} exists **then**
3: **return** "Already exists;" // Patient Record associated with P_{ID} already exists
4: **end if**
5: $MSPID \leftarrow$ GetMSPID(ctx) // Retrieve Membership Service Provider ID
6: **if** IsForbiddenToCreateEhr($MSPID$) **then**
7: **return** "Forbidden access;"
8: **end if**
9: $cid \leftarrow$ GetClientIdentity() // Retrieve client identity to determine user attributes
10: **if** $notcid.assertAttributeValue('userattrib.writer','true')$ **then**
11: **return** "You are not allowed to create or update an asset"
12: **else**
13: **INITIALIZE** algorithm, $securityKey$, $initVector$
14: $Enc_{rypt}(Ehr)$ // Encrypt Ehr
15: $Ehr \leftarrow P_{name}, DOB, Addr, PhNo, Ins, Med, Diag, Test_{Tn}$
16: Store the encrypted data on the storage
17: **end if**

In the CreateEhr() function, patients maintain ownership of their records linked to their unique ID. The records include information about the consulted healthcare provider (doctor's ID) and details related to pharmacy transactions (pharmacy record ID). The ID is

Algorithm 7.2 Pseudocode of ReadEhr function

Require: P_{ID}
Ensure: Patient Record Retrieved
Steps:
// Check if the user's MSPID is allowed to retrieve information
if not allowedToRetrieveInformation(MSPID) **then**
    return "Forbidden access"
end if
// Check if the user has the necessary attributes for access
if not hasAttribute(cid, allowedAttributes) **then**
    return "Do not have the privilege to read record"
else
    // Retrieve a record based on the patient ID
    // Decrypt sensitive attributes in the EHR record
end if

Algorithm 7.3 Pseudocode of UpdateEhr function

Require: P_{ID}, Med, T_{status}
Ensure: Patient Record Updated
Steps:
// Check if the user's MSPID is allowed to update information
$MSPID \leftarrow$ GetMSPID(ctx)
if IsForbiddenToUpdateEhr($MSPID$) **then**
    return "Forbidden access"
end if
// Check if the user has the necessary attributes for access
if not hasAttribute(cid, allowedAttributes) **then**
    return "Do not have the privilege to update record"
else
    // Authorized user can update intended record.
    // Update Ehr with modified fields: Med, T_{status}
    // Store the updated record on the storage
end if

key for authorized access to the respective records by the designated record owners. Additionally, patients can monitor the status of their insurance claims, specifically regarding the approval or denial of their insurance coverage, depending on the permissions provided by the insurance entity, as demonstrated in algorithm 7.1, algorithm 7.2, and algorithm 7.3.

The Doctor entity (Algorithm 7.4) maintains patient IDs for individuals requesting a medical diagnosis from the corresponding doctor. The Doctor entity must be authorized to access comprehensive or specific patient information. ABAC efficiently governs data access control in our system. In compliance with the GDPR and HIPAA regulatory mandates, patients must be granted full control over their EHRs. It includes allowing patients to revoke access

Algorithm 7.4 Pseudocode of CreateDoctorRecord function

Require: $D_{ID}, P_{ID}, D_{name}, Gender, D_{Spec}, D_{Work}$
Ensure: Doctor Record Created

```
Steps:
if D_ID exists then
    return "Already exists;" // Record already exists
end if
// Retrieve the client identity (CID) to determine user attributes
// Check the user's MSPID to determine their role
// Retrieve the client identity to determine user attributes
// Check if the user has the intended attribute;
if not cid.assertAttributeValue('userattrib.writer', 'true') then
    return "Access denied to create records"
else
    // Encrypt sensitive attributes
    // Create a new record with the provided data.
    // Store the encrypted record on the storage.
end if
```

privileges for viewing their EHRs and to request the deletion of their EHRs.

Deployment Phase : The proposed medical data record system is based on the blockchain-based Hyperledger Fabric framework, part of the Hyperledger project. Hyperledger represents an open-source DLT functioning within a permissioned framework. Created by the Linux Foundation, its main objective is to enable the execution of various smart contracts and the deployment of multiple applications within a blockchain network. Within the framework of HLF, smart contracts are deployed, executed, and evaluated using tools like Hyperledger Caliper. Additionally, it is crucial to note that HLF is not confined to a particular domain, as it supports multiple programming languages, including Java, Go, Node.js, and others, facilitating the creation of contracts and business networks across a diverse range of applications.

7.7 Performance Evaluation

7.7.1 Feature Comparison

This study provides a theoretical analysis of our research, comparing it with other blockchain healthcare solutions. Table 7.2 compares our proposed healthcare solution utilizing blockchain technology against existing solutions in the healthcare sector. ✓ indicates that the solution includes feature while ✗ signifies that the feature is absent from the solution. The research

Table 7.2: Comparison of our proposed solution with existing solutions leveraging blockchain in healthcare

Ref.	$Param_1$	$Param_2$	$Param_3$	$Param_4$	$Param_5$	$Param_6$	$Param_7$	$Param_8$
[168]	✓	✗	✓	✓	✗	✗	✗	✗
[169]	✗	✗	✓	✓	✓	✗	✗	✗
[170]	✗	✗	✓	✗	✓	✗	✗	✗
[171]	✗	✗	✓	✓	✓	✗	✓	✓
[63]	✗	✗	✓	✓	✓	✗	✓	✓
[57]	✗	✗	✓	✓	✓	✓	✓	✓
[172]	✓	✓	✓	✓	✓	✗	✗	✗
[53]	✗	✗	✓	✓	✓	✗	✗	✗
[84]	✗	✗	✓	✓	✗	✗	✓	✗
[173]	✗	✓	✓	✗	✓	✗	✗	✗
[174]	✗	✗	✓	✓	✓	✗	✗	✗
[175]	✗	✗	✓	✗	✓	✗	✗	✗
ABHealChain	✓	✓	✓	✓	✓	✓	✓	✓

$Param_1$: Access control, $Param_2$: Secure data sharing, $Param_3$: Privacy, $Param_4$: Data security, $Param_5$: Integrity, $Param_6$: User authentication, $Param_7$: Scalability, $Param_8$: Distributed file storage

demonstrates that our solution effectively addresses the requirements of a secure healthcare system in the following areas: controlling access, securely sharing data, ensuring privacy and data security, authenticating users, scalability, and establishing a distributed file system.

7.7.2 Experimental Analysis

The proposed approach is simulated using a personal computer device. In the simulation environment, designated components are configured, comprising an orderer, five organizations, and the creation of communication channels, as depicted in Table 7.3. Each organization possesses a peer, referred to as $peer0\langle ORG\rangle$. To evaluate the usefulness and efficiency of this system, we have employed Hyperledger Caliper version 0.6.0. This specialized tool evaluates the performance of a blockchain network installation, capturing crucial performance indicators for a full analysis.

The Caliper [118] presents an array of performance measures, encompassing throughput, success rate, latency, traffic in, traffic out, CPU utilization, and memory utilization Table 7.4. CPU use is a measure that indicates the extent to which a computer or server's CPU is being employed at a specific time. CPU utilization aids in comprehending system load, identifying performance bottlenecks, optimizing resource allocation, and facilitating

Table 7.3: Experimental setup and configuration for HLF performance evaluation

Component	**Parameters**
Operating System	Ubuntu 22.04 LTS
Hyperledger Fabric	v2.5.0
Benchmarking tool	Hyperledger Caliper, version 0.6.0
Chaincode language	JavaScript
Off-chain storage	IPFS
Transaction Type	Create, Read, Update
Transaction rate	50, 75, 100, 150, 200, 250
Number of transactions	10, 200, 400, 600, 800, 1000
Number of Worker	5, 25, 50
Number of Organizations	5
Channel	Single private channel

capacity planning. The system's performance has been assessed by systematically altering

Table 7.4: Measuring the resource consumption for 1000 transactions

Name	**CPU% (max)**	**CPU% (avg)**	**Memory (max)[MB]**	**Memory (avg)[MB]**	**Traffic In[MB]**	**Traffic Out[MB]**	**Disc Write[MB]**	**Disc Read[B]**
(Resource Utilization for a Create function)								
/peer0.epharma.example.com	6.08	2.19	128	123	7.82	2.87	8.27	0.00
/peer0.doctor.example.com	5.68	2.08	127	123	7.79	2.81	8.27	0.00
/peer0.patient.example.com	6.18	2.22	159	151	7.81	3.40	8.27	0.00
/peer0.pathology.example.com	5.58	2.04	131	128	7.76	2.77	8.27	24.0
/peer0.insurance.example.com	6.11	2.19	125	122	7.87	2.92	8.27	0.00
/orderer.example.com	2.12	0.71	82.0	71.2	6.48	30.4	12.9	0.00
(Resource Utilization for a Read function)								
/peer0.epharma.example.com	0.16	0.14	127	127	0.0956	0.279	0.00	0.00
/peer0.doctor.example.com	0.15	0.14	127	126	0.0891	0.276	0.00	0.00
/peer0.patient.example.com	3.81	1.39	159	158	2.45	4.05	0.00	0.00
/peer0.pathology.example.com	0.2	0.16	131	131	0.0945	0.282	0.00	24.0
/peer0.insurance.example.com	0.24	0.16	125	125	0.0933	0.283	0.00	0.00
/orderer.example.com	0.01	0.01	82.1	82.1	0.0440	0.0640	0.00	0.00
(Resource Utilization for an Update function)								
/peer0.epharma.example.com	6.38	2.29	134	132	7.61	2.81	8.14	0.00
/peer0.doctor.example.com	5.54	2	132	130	7.61	2.80	8.14	0.00
/peer0.patient.example.com	6.15	2.19	171	166	7.64	3.39	8.14	32.0
/peer0.pathology.example.com	6	2.16	138	135	7.57	2.74	8.14	0.00
/peer0.insurance.example.com	6.03	2.15	132	130	7.58	2.76	8.14	0.00
/orderer.example.com	2.07	0.69	96.1	81.8	6.38	29.7	12.7	0.00

the transaction amount while sustaining a constant transaction rate. The number of transactions will be varied at 10, 200, 400, 600, 800, and 1000 while maintaining a continuous trans-

action rate of 150 tps to evaluate the effect of transaction volume on blockchain throughput and latency. For CreateEhr(), CreateDoctorRecord(), CreatePharmacyRecord(), CreateLabRecord(), and CreateInsuranceRecord(), 1000 transactions are executed at a rate of 150 tps, as illustrated in Figure 7.7. The recorded throughput is 146.5 tps, 114.7 tps, 114.6 tps, 114.6 tps, and 114.7 tps, while the average latency is 0.90 seconds, 0.13 seconds, 0.10 seconds, 0.11 seconds, and 0.11 seconds, respectively, as shown in Figure 7.7. Additionally,

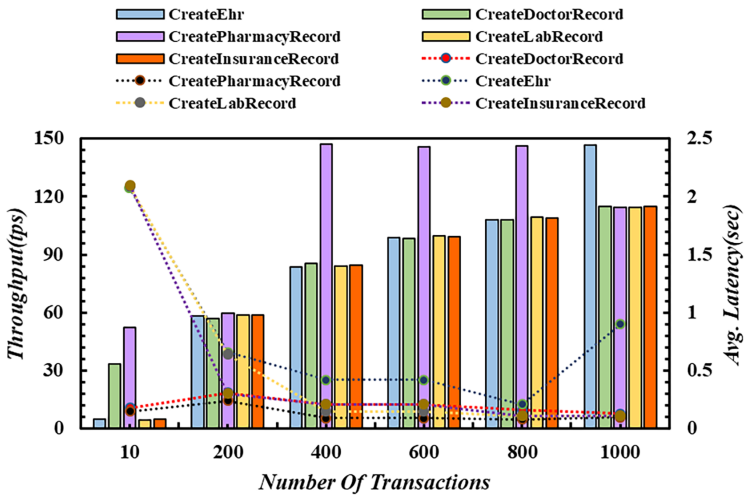


Figure 7.7: Average latency and throughput measures in creating the patient, doctor, pharmacy, insurance, and path lab record

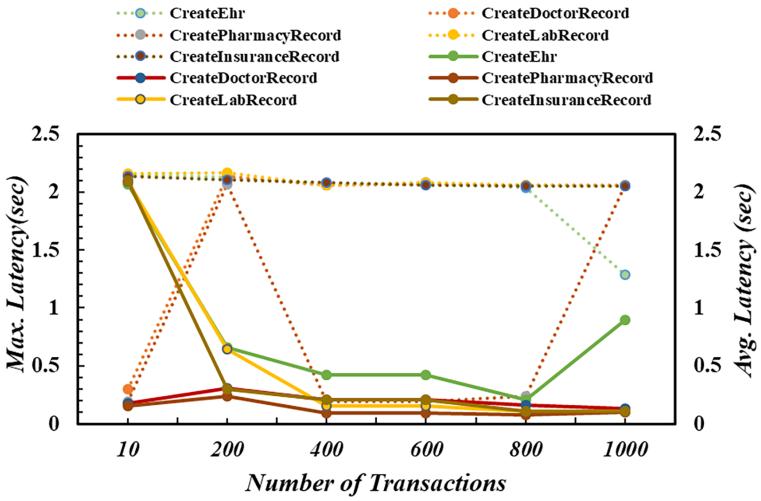


Figure 7.8: Maximum latency and average latency measures in creating patient, doctor, pharmacy, insurance, and path lab record

maximum and average latency are presented in Figure 7.8. Our trials revealed a decline in throughput following a specific number of transactions and at certain transaction rates. This observation indicates that as the system undergoes an increase in load beyond a par-

ticular threshold, it may face bottlenecks or performance constraints. Similarly, Figure 7.9

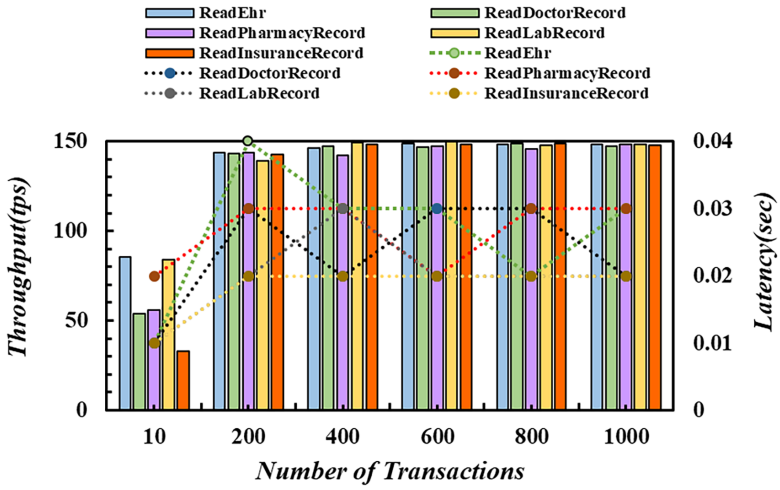


Figure 7.9: Latency and throughput measures in retrieving the patient, doctor, pharmacy, insurance, and path lab record

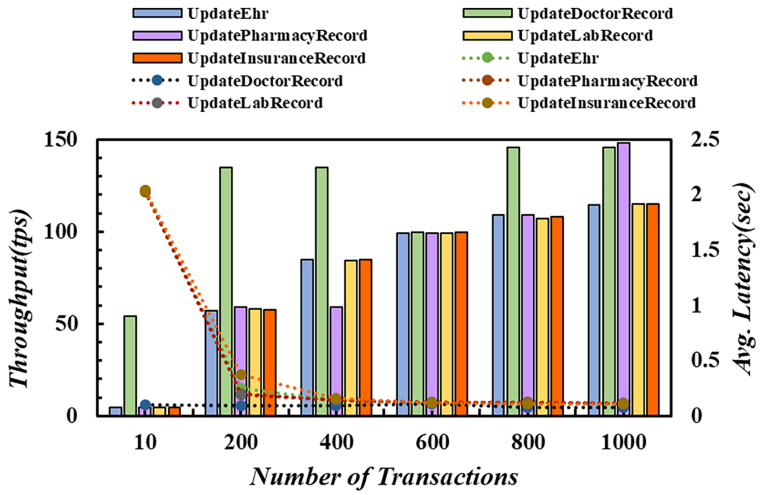


Figure 7.10: Average latency and throughput measures in updating the patient, doctor, pharmacy, insurance, and path lab record

illustrates the throughput and latency for ReadEhr(), ReadDoctorRecord(), ReadPharmacyRecord(), ReadLabRecord() and ReadInsuranceRecord(). The number of transactions sent in blockchain are 10, 200, 400, 600, 800, and 1000 at the rate of 150 tps. In the graph, maximum latency is calculated for 1000 transactions and calculated max. latency is 0.03 sec, 0.02 sec, 0.03 sec, 0.02 sec, and 0.02 sec. The avg. latency calculated for txNumber 10, 200, 4000, 600, 800, and 1000 is 0.01 sec. The observed latency in all the cases are 0.03 sec and 0.02 sec which is slightly more than the average latency. Within the throughput

presented in Figure 7.9, the observed values for transactions 200, 400, 600, 800, and 1000 are quite close, with negligible differences among them.

Likewise, Figure 7.10 also illustrates the average delay and throughput during record updates. An increase in latency is seen when the transaction count is fixed at 10, with a transaction rate of 150 transactions per second (tps). Moreover, the findings indicate that as the transaction count escalates to 200, 400, 600, 800, and 1000, there is a significant decrease in latency relative to a situation with merely 10 transactions (txNumber 10).

Furthermore, Figure 7.11 illustrates the maximum and average latency metrics associated with updating patient, doctor, pharmacy, insurance, and pathology medical records. A

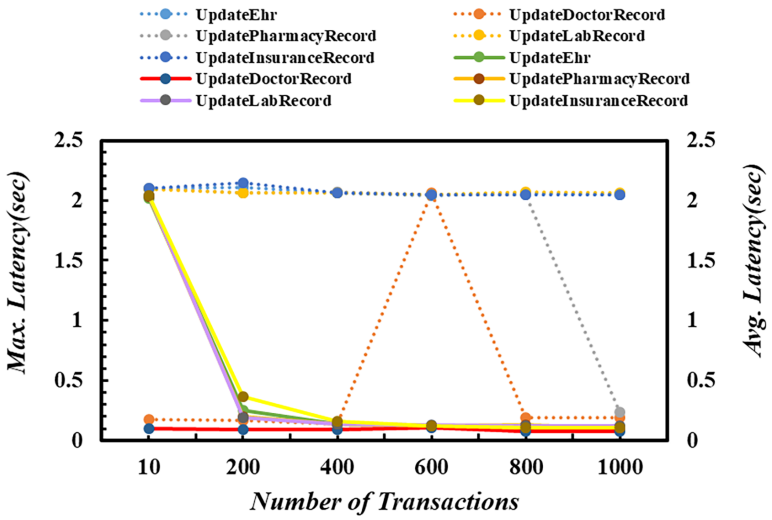


Figure 7.11: Maximum, average latency measures in updating the patient, doctor, pharmacy, insurance, and path lab record

comparative analysis of latency and throughput is conducted in by varying tps and the number of workers. We calculated the average latency for viewing records in our system by assessing the latency across various record types, including EHR, Doctor, Pharma, Lab, and Insurance. The analysis is performed at configured send rates of 50, 100, 150, 200, and 250 tps. The average latency from these assessments are combined to determine the overall average latency for record viewing in the system. The average latency of 0.01 seconds is consistent across 50, 100, 150, 200, and 250 transactions per second, as shown in Figure 7.12 and Figure 7.13.

Likewise, Figure 7.14 displays the mean latency associated with updating health records. The average latencies at the configured send rates of 50, 100, 150, 200, and 250 tps are 0.136 seconds, 0.090 seconds, 0.076 seconds, 0.116 seconds, and 0.178 seconds, respectively. The

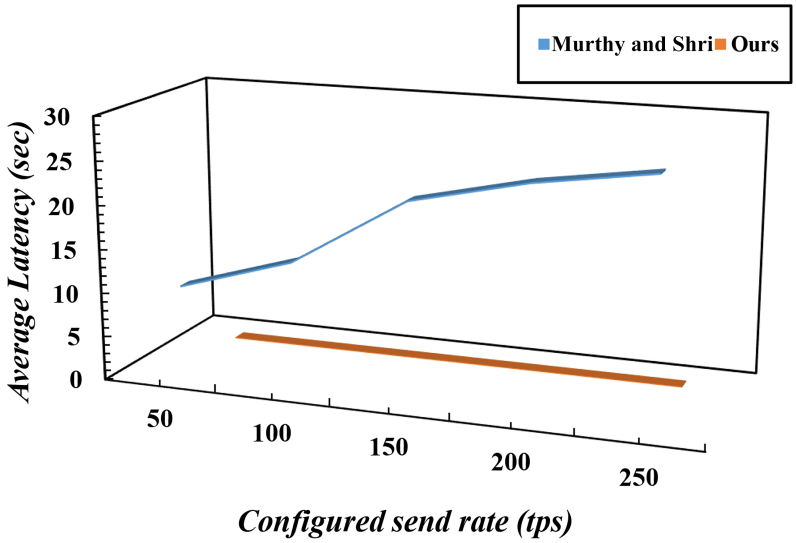


Figure 7.12: Average latency to view the record

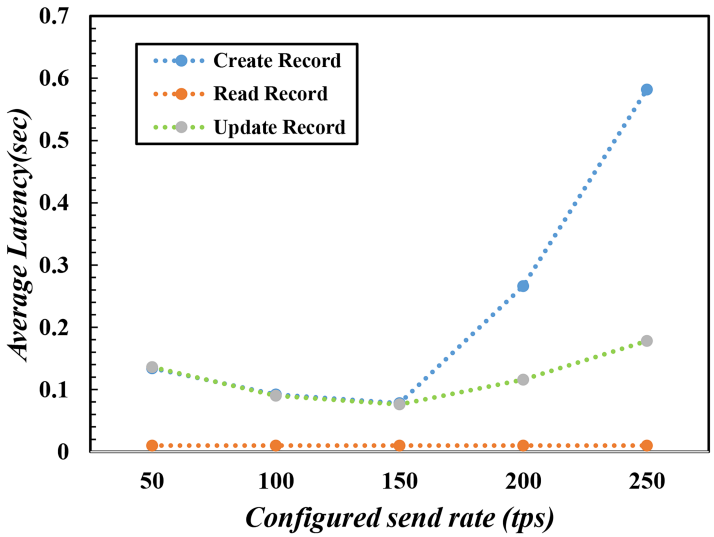


Figure 7.13: Average latency to create, read, and update the record

values are lower than those reported in ([176]).

The throughput observed for record creation with 5, 25, and 50 workers processing 1,000 transactions at 75 tps is 75.1, 75.6, and 76.4 tps, respectively, according to [177]. In contrast, our system achieved 75.1, 74, and 70 tps, suggesting that the throughput reported in [177] is marginally superior (Figure 7.15). In contrast, the latency measured for our system with 5, 25, and 50 workers is 0.1, 0.11, and 0.134 seconds, respectively, which is lower than the latency reported by [177], which are 0.1, 0.11, and 0.76 seconds.

The ABHealChain system uses transaction latency and throughput as its primary metrics to assess scalability. It has been demonstrated through experiments that the system can

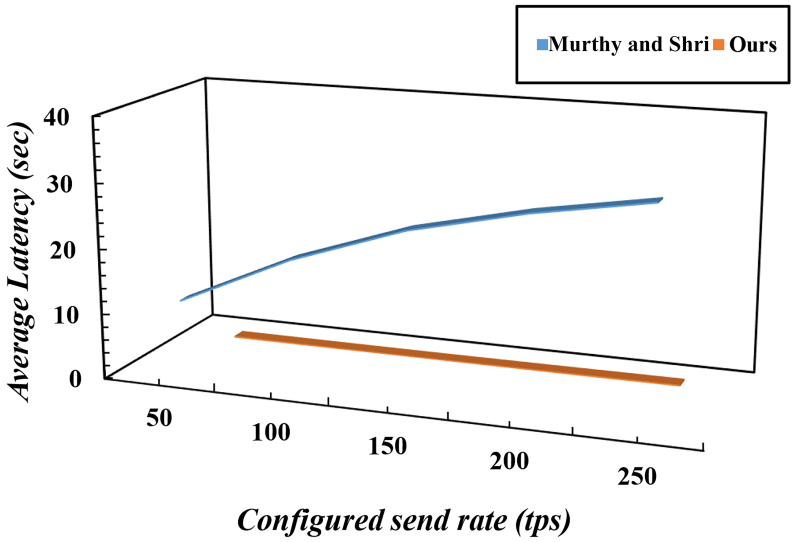


Figure 7.14: Average latency to update the record

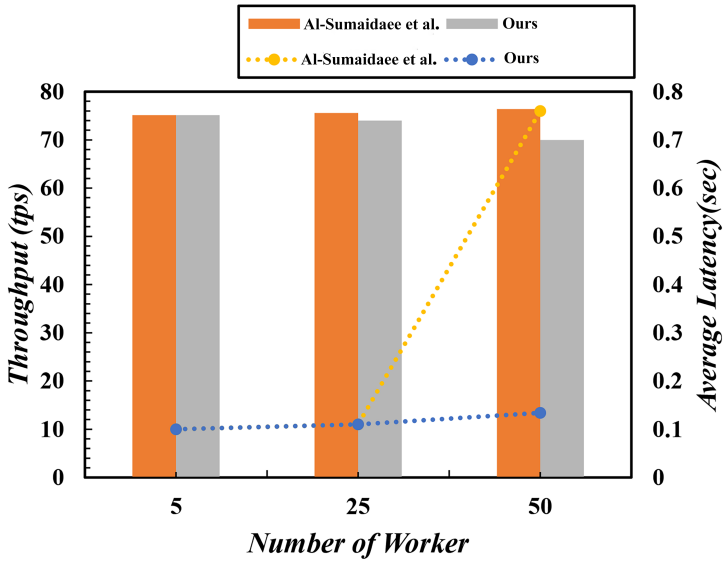


Figure 7.15: Average latency and throughput to create the record

manage an increasing workload without experiencing a decrease in performance. In addition, the system's latency stays the same or hardly increases marginally as the demand increases. Increasing the send rate or the quantity of transactions does not affect the system's constant response times.

7.8 Summary

This study utilized HLF technology to develop the ABHealChain framework. The proposed solution employs HLF as the foundational technology to address the identified issues in the

healthcare sector's methodologies and data management systems. Various stakeholders are engaged in the selected use case, including the patient, physician, laboratory, pharmacy, and insurance provider. Peer nodes interact through chaincode to securely store data in the ledger.

This research aims to improve healthcare data management, security, and confidentiality by applying HLF and advanced cryptographic protocols. The information is stored in blocks, thereby addressing security concerns effectively. IPFS has been integrated to enhance data accessibility. An encryption scheme and access control mechanism have been established to protect sensitive medical data from unauthorized access and prevent intentional and inadvertent alterations.

This paper addresses the challenges associated with access control, data integrity, data sharing, and data storage. Integrating IoT devices is expected to generate significant amounts of data in the upcoming phases of healthcare system advancement. In this context, we propose utilizing HLF as a functional tool, providing a secure and transparent data management and analysis framework.

The final chapter of this thesis concludes the research and outlines potential directions for future work. It highlights key contributions and suggests avenues for further investigation, building upon the findings presented in this thesis.

Chapter 8

Conclusion and Future Directions

8.1 Conclusion and Future Directions

In recent years, advancements in blockchain technology have revolutionized various sectors, particularly healthcare, by providing innovative solutions for secure data management, enhanced privacy, and improved operational efficiency. There are many types of blockchain, each having their own advantage and shortcoming. This thesis investigates the application of HLF blockchain technology to enhance security, privacy, and performance in healthcare data management. By analyzing scalability and performance of HLF, Ethereum and Hyperledger Besu, the research demonstrates HLF effectively handles large transaction volumes with minimal latency. The study highlights the superior performance of the Raft ordering service over Solo and identifies factors affecting scalability, including network size and resource capacity. To address healthcare-specific privacy challenges, the BloCPABE scheme is proposed, providing fine-grained, efficient access control with reduced computational costs and strong resistance to collusion attacks. Additionally, the SmarConTest methodology improves the detection of vulnerabilities in Ethereum smart contracts, achieving high accuracy and fast prediction. The ABHealChain framework integrates HLF with IPFS and advanced cryptographic techniques to secure and streamline healthcare data exchange among diverse stakeholders. Overall, this thesis underscores the potential of blockchain and cryptographic solutions to transform healthcare data management by ensuring robust security, privacy, and scalability. The future work may include:

[1] We will investigate the impact of the endorsement policy on HLF.

[2] We aim to examine the impact of the endorsement policy on HLF.

[3] We plan to analyze various consensus protocols used in permissioned blockchain systems.

[4] We intend to explore solutions to resolve the MVCC_READ_CONFLICT error in blockchain transactions.

[5] We aim to investigate different deep learning models and feature extraction methods for detecting vulnerabilities in smart contracts.

[6] We plan to curate a comprehensive labeled dataset using various static and dynamic analysis tools, covering vulnerabilities in Ethereum smart contracts categorized as critical, high, and medium.

[7] We propose the development of a system for the online detection of vulnerabilities in Ethereum smart contracts.

[8] We aim to explore the use of multimodal input components such as source code, bytecode, and abstract syntax tree for vulnerability detection.

[9] We plan to utilize graph neural networks to capture complex relationships within the data to enhance detection performance.

[10] We aim to develop a scheme that integrates attribute-based encryption with proxy re-encryption for enhanced data security.

[11] We plan to explore the use of composite order groups in the development of attribute-based encryption schemes.

[12] Instead of relying on fog computing to reduce decryption time, we will investigate techniques that allow for message decryption in constant time.

[13] We aim to design and develop an efficient consensus algorithm that is fault-tolerant, reaches consensus quickly, is scalable, and is energy efficient.

List of Publications

Refereed journals:

[1] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Workload dynamics implications in permissioned blockchain scalability and performance," *Cluster Computing*, Springer, vol. 27, issue. 8, pp.11569-11593, 2024, DOI: https://doi.org/10.1007/s10586-024-04550-z. **(SCIE, IF=3.6)**

[2] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Exploring the Transformative Impact of Blockchain Technology on Healthcare: Security, Challenges, Benefits, and Future Outlook," *Transactions on Emerging Telecommunications Technologies* , Wiley, vol. 36, p. e70087, 2025, DOI: https://doi.org/10.1002/ett.70087. **(SCIE, IF=2.5)**.

[3] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Revocable and Privacy-Preserving CP-ABE Scheme for Secure mHealth Data Access in Blockchain," *Concurrency and Computation: Practice and Experience*, Wiley, vol. 37, p. e70064, 2025, DOI: https://doi.org/10.1002/cpe.70064. **(SCIE, IF=1.5)**.

[4] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "ABHealChain: Enhancing Privacy and Security in Healthcare Data Sharing through Hyperledger Fabric and Attribute-based Access Control", –communicated

[5] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "SmarConTest: An Efficient Smart Contract Vulnerability Detection Method Using Machine Learning," —communicated

International conferences:

[1] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Performance benchmarking and analysis of blockchain platforms", *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, Delhi, pp. 1-7.

[2] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Attribute-based encryption scheme for secure and efficient access in blockchain", *2024 IEEE International Con-*

ference for Women in Innovation, Technology & Entrepreneurship (ICWITE), Bangalore, pp. 653-658.

[3] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "Simulation-based Performance Evaluation of Consensus Algorithms in NS3 for Blockchain Network", *International Conference on Futuristic Technologies (INCOFT2025)*, Pune.

[4] **Anita Thakur**, Virender Ranga and Ritu Agarwal, "A Review On Smart Contract Vulnerability Detection Using Deep Learning", —communicated

Bibliography

[1] I. Bashir, *Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more.* Packt Publishing Ltd, 2020.

[2] S. Haber and W. S. Stornetta, *How to time-stamp a digital document.* Springer, 1991.

[3] S. Nakamoto, "A peer-to-peer electronic cash system," *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, vol. 4, no. 2, p. 15, 2008.

[4] J. Xu, C. Wang, and X. Jia, "A survey of blockchain consensus protocols," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–35, 2023. https://doi.org/10.1145/3579845.

[5] S. Ramzan, A. Aqdus, V. Ravi, D. Koundal, R. Amin, and M. A. Al Ghamdi, "Healthcare applications using blockchain technology: Motivations and challenges," *IEEE Transactions on Engineering Management*, vol. 70, no. 8, pp. 2874–2890, 2022. 10.1109/TEM.2022.3189734.

[6] A. Thakur, V. Ranga, and R. Agarwal, "Attribute-based encryption scheme for secure and efficient access in blockchain," in *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, pp. 653–658, IEEE, 2024. 10.1109/ICWITE59797.2024.10502721.

[7] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, "Performance analysis of hyperledger fabric platforms," *Security and Communication Networks*, vol. 2018, no. 1, p. 3976093, 2018. https://doi.org/10.1155/2018/3976093.

[8] S. Shalaby, A. A. Abdellatif, A. Al-Ali, A. Mohamed, A. Erbad, and M. Guizani, "Performance evaluation of hyperledger fabric," in *2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT)*, pp. 608–613, IEEE, 2020. 10.1109/ICIoT48696.2020.9089614.

[9] T. Nakaike, Q. Zhang, Y. Ueda, T. Inagaki, and M. Ohara, "Hyperledger fabric performance characterization and optimization using goleveldb benchmark," in *2020*

IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 1–9, IEEE, 2020. https://doi.org/10.1109/ICBC48266.2020.9169454.

[10] J. Dreyer, M. Fischer, and R. Tönjes, "Performance analysis of hyperledger fabric 2.0 blockchain platform," in *Proceedings of the workshop on cloud continuum services for smart IoT systems*, pp. 32–38, 2020. https://doi.org/10.1145/3417310.3431398.

[11] C. Melo, F. Oliveira, J. Dantas, J. Araujo, P. Pereira, R. Maciel, and P. Maciel, "Performance and availability evaluation of the blockchain platform hyperledger fabric," *The Journal of Supercomputing*, vol. 78, no. 10, pp. 12505–12527, 2022. https://doi.org/10.1007/s11227-022-04361-2.

[12] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, "Performance analysis of private blockchain platforms in varying workloads," in *2017 26th international conference on computer communication and networks (ICCCN)*, pp. 1–6, IEEE, 2017. https://doi.org/10.1109/ICCCN.2017.8038517.

[13] C. Wang and X. Chu, "Performance characterization and bottleneck analysis of hyperledger fabric," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1281–1286, IEEE, 2020. https://doi.org/10.1109/ICDCS47774.2020.00165.

[14] S. Mollajafari and K. Bechkoum, "Blockchain technology and related security risks: Towards a seven-layer perspective and taxonomy," *Sustainability*, vol. 15, no. 18, p. 13401, 2023. https://doi.org/10.3390/su151813401.

[15] L. Duan, Y. Sun, K. Zhang, and Y. Ding, "Multiple-layer security threats on the ethereum blockchain and their countermeasures," *Security and Communication Networks*, vol. 2022, no. 1, p. 5307697, 2022. https://doi.org/10.1155/2022/5307697.

[16] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *Ieee Access*, vol. 7, pp. 164908–164940, 2019. https://doi.org/10.1109/ACCESS.2019.2950872.

[17] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE network*, vol. 33, no. 5, pp. 166–173, 2019. https://doi.org/10.1109/MNET.001.1800290.

[18] S. Tu, M. Waqas, F. Huang, G. Abbas, and Z. H. Abbas, "A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing," *Computer Networks*, vol. 195, p. 108196, 2021. https://doi.org/10.1016/j.comnet.2021.108196.

[19] Y. Miao, F. Li, X. Li, J. Ning, H. Li, K.-K. R. Choo, and R. H. Deng, "Verifiable outsourced attribute-based encryption scheme for cloud-assisted mobile e-health system," *IEEE Transactions on Dependable and Secure Computing*, 2023. https://doi.org/10.1109/TDSC.2023.3292129.

[20] A. Thakur, V. Ranga, and R. Agarwal, "Revocable and privacy-preserving cp-abe scheme for secure mhealth data access in blockchain," *Concurrency and Computation: Practice and Experience*, vol. 37, no. 9-11, p. e70064, 2025. https://doi.org/10.1002/cpe.70064.

[21] P. Qian, Z. Liu, Q. He, B. Huang, D. Tian, and X. Wang, "Smart contract vulnerability detection technique: A survey," *arXiv preprint arXiv:2209.05872*, 2022. https://doi.org/10.13328/j.cnki.jos.006375.

[22] Z. Liao, Z. Zheng, X. Chen, and Y. Nan, "Smartdagger: a bytecode-based static analysis approach for detecting cross-contract vulnerability," in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 752–764, 2022. https://doi.org/10.1145/3533767.3534222.

[23] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, and Y. Alexandrov, "Smartcheck: Static analysis of ethereum smart contracts," in *Proceedings of the 1st international workshop on emerging trends in software engineering for blockchain*, pp. 9–16, 2018. https://doi.org/10.1145/3194113.3194115.

[24] S. Wang and X. Zhao, "Contractsentry: a static analysis tool for smart contract vulnerability detection," *Automated Software Engineering*, vol. 32, no. 1, p. 1, 2025. https://doi.org/10.1007/s10515-024-00471-8.

[25] H. Chen, X. Zhao, Y. Wang, and Z. Zhen, "Safecheck: Detecting smart contract vulnerabilities based on static program analysis methods," *Security and Privacy*, vol. 7, no. 5, p. e393, 2024. https://doi.org/10.1002/spy2.393.

[26] N. Ashizawa, N. Yanai, J. P. Cruz, and S. Okamura, "Eth2vec: learning contract-wide code representations for vulnerability detection on ethereum smart contracts,"

in *Proceedings of the 3rd ACM international symposium on blockchain and secure critical infrastructure*, pp. 47–59, 2021. https://doi.org/10.1145/3457337.3457841.

[27] A. Ghaleb, J. Rubin, and K. Pattabiraman, "etainter: detecting gas-related vulnerabilities in smart contracts," in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 728–739, 2022. https://doi.org/10.1145/3533767.3534378.

[28] J. Chen, X. Xia, D. Lo, J. Grundy, X. Luo, and T. Chen, "Defectchecker: Automated smart contract defect detection by analyzing evm bytecode," *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2189–2207, 2021. https://doi.org/10.1109/TSE.2021.3054928.

[29] F. Contro, M. Crosara, M. Ceccato, and M. Dalla Preda, "Ethersolve: Computing an accurate control-flow graph from ethereum bytecode," in *2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC)*, pp. 127–137, IEEE, 2021. https://doi.org/10.1109/ICPC52881.2021.00021.

[30] T. D. Nguyen, L. H. Pham, and J. Sun, "Sguard: towards fixing vulnerable smart contracts automatically," in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 1215–1229, IEEE, 2021. https://doi.org/10.1109/SP40001.2021.00057.

[31] X. Tang, Y. Du, A. Lai, Z. Zhang, and L. Shi, "Deep learning-based solution for smart contract vulnerabilities detection," *Scientific Reports*, vol. 13, no. 1, p. 20106, 2023. https://doi.org/10.1038/s41598-023-47219-0.

[32] Z. Liu, P. Qian, X. Wang, Y. Zhuang, L. Qiu, and X. Wang, "Combining graph neural networks with expert knowledge for smart contract vulnerability detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1296–1310, 2023. 10.1109/TKDE.2021.3095196.

[33] P. T. Duy, N. H. Khoa, N. H. Quyen, L. C. Trinh, V. T. Kien, T. M. Hoang, and V.-H. Pham, "Vulnsense: efficient vulnerability detection in ethereum smart contracts by multimodal learning with graph neural network and language model," *International Journal of Information Security*, vol. 24, no. 1, p. 48, 2025. https://doi.org/10.1007/s10207-024-00965-2.

[34] W. Deng, H. Wei, T. Huang, C. Cao, Y. Peng, and X. Hu, "Smart contract vulnerability detection based on deep learning and multimodal decision fusion," *Sensors*, vol. 23, no. 16, p. 7246, 2023. https://doi.org/10.3390/s23167246.

[35] H. Chu, P. Zhang, H. Dong, Y. Xiao, and S. Ji, "Deepfusion: Smart contract vulnerability detection via deep learning and data fusion," *IEEE Transactions on Reliability*, 2024. https://doi.org/10.1109/TR.2024.3480010.

[36] P. Qian, Z. Liu, Y. Yin, and Q. He, "Cross-modality mutual learning for enhancing smart contract vulnerability detection on bytecode," in *Proceedings of the ACM Web Conference 2023*, pp. 2220–2229, 2023. https://doi.org/10.1145/3543507.3583367.

[37] Y. Liu, C. Wang, and Y. Ma, "Dl4sc: a novel deep learning-based vulnerability detection framework for smart contracts," *Automated Software Engineering*, vol. 31, no. 1, p. 24, 2024. https://doi.org/10.1007/s10515-024-00418-z.

[38] S.-J. Hwang, S.-H. Choi, J. Shin, and Y.-H. Choi, "Codenet: Code-targeted convolutional neural network architecture for smart contract vulnerability detection," *IEEE Access*, vol. 10, pp. 32595–32607, 2022. https://doi.org/10.1109/ACCESS.2022.3162065.

[39] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, the Netherlands, February 21-24, 2007. Proceedings 4*, pp. 515–534, Springer, 2007. https://doi.org/10.1007/978-3-540-70936-7_28.

[40] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 121–130, 2009. https://doi.org/10.1145/1653662.1653678.

[41] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Annual international conference on the theory and applications of cryptographic techniques*, pp. 568–588, Springer, 2011. https://doi.org/10.1007/978-3-642-20465-4_31.

[42] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," *IEEE transactions on parallel and distributed systems*,

vol. 25, no. 7, pp. 1735–1744, 2013. https://doi.org/10.1109/TPDS.2013.253.

[43] Y. Guo, Z. Lu, H. Ge, and J. Li, "Revocable blockchain-aided attribute-based encryption with escrow-free in cloud storage," *IEEE Transactions on Computers*, vol. 72, no. 7, pp. 1901–1912, 2023. https://doi.org/10.1109/TC.2023.3234210.

[44] J. Wei, X. Chen, X. Huang, X. Hu, and W. Susilo, "Rs-habe: Revocable-storage and hierarchical attribute-based access scheme for secure sharing of e-health records in public cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2301–2315, 2019. https://doi.org/10.1109/TDSC.2019.2947920.

[45] Z. Hou, J. Ning, X. Huang, S. Xu, and L. Y. Zhang, "Blockchain-based efficient verifiable outsourced attribute-based encryption in cloud," *Computer Standards & Interfaces*, vol. 90, p. 103854, 2024. https://doi.org/10.1016/j.csi.2024.103854.

[46] B. Zhang, W. Yang, F. Zhang, and J. Ning, "Efficient attribute-based searchable encryption with policy hiding over personal health records," *IEEE Transactions on Dependable and Secure Computing*, 2024. https://doi.org/10.1109/TDSC.2024.3432769.

[47] X. Liang, Y. Liu, and J. Ning, "An access control scheme with privacy-preserving authentication and flexible revocation for smart healthcare," *IEEE Journal of Biomedical and Health Informatics*, 2024. https://doi.org/10.1109/JBHI.2024.3391218.

[48] B. K. Rai, "Pcbehr: patient-controlled blockchain enabled electronic health records for healthcare 4.0," *Health Services and Outcomes Research Methodology*, vol. 23, no. 1, pp. 80–102, 2023. https://doi.org/10.1007/s10742-022-00279-7.

[49] A. Ekblaw and A. Azaria, "Medrec: Medical data management on the blockchain," *Viral Communications*, 2016.

[50] Y. Chen, S. Ding, Z. Xu, H. Zheng, and S. Yang, "Blockchain-based medical records secure storage and medical service framework," *Journal of medical systems*, vol. 43, pp. 1–9, 2019. https://doi.org/10.1007/s10916-018-1121-4.

[51] M. A. Saberi, M. Adda, and H. Mcheick, "Break-glass conceptual model for distributed ehr management system based on blockchain, ipfs and abac," *Procedia Computer Sci-*

ence, vol. 198, pp. 185–192, 2022. https://doi.org/10.1016/j.procs.2021.12.227.

[52] A. Dubovitskaya, F. Baig, Z. Xu, R. Shukla, P. S. Zambani, A. Swaminathan, M. M. Jahangir, K. Chowdhry, R. Lachhani, N. Idnani, *et al.*, "Action-ehr: Patient-centric blockchain-based electronic health record data management for cancer care," *Journal of medical Internet research*, vol. 22, no. 8, p. e13598, 2020. https://doi.org/10.2196/13598.

[53] U. Chelladurai and S. Pandian, "A novel blockchain based electronic health record automation system for healthcare," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 693–703, 2022. https://doi.org/10.1007/s12652-021-03163-3.

[54] P. P. Ray, B. Chowhan, N. Kumar, and A. Almogren, "Biothr: Electronic health record servicing scheme in iot-blockchain ecosystem," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10857–10872, 2021. https://doi.org/10.1109/JIOT.2021.3050703.

[55] A. Farouk, A. Alahmadi, S. Ghose, and A. Mashatan, "Blockchain platform for industrial healthcare: Vision and future opportunities," *Computer Communications*, vol. 154, pp. 223–235, 2020. https://doi.org/10.1016/j.comcom.2020.02.058.

[56] K. B. Adsul and S. Kosbatwar, "A novel approach for traceability & detection of counterfeit medicines through blockchain," *International Journal of Current Engineering and Technology*, 2020.

[57] M. Uddin, "Blockchain medledger: Hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry," *International Journal of Pharmaceutics*, vol. 597, p. 120235, 2021. https://doi.org/10.1016/j.ijpharm.2021.120235.

[58] N. Saxena, I. Thomas, P. Gope, P. Burnap, and N. Kumar, "Pharmacrypt: Blockchain for critical pharmaceutical industry to counterfeit drugs," *Computer*, vol. 53, no. 7, pp. 29–44, 2020. https://doi.org/10.1109/MC.2020.2989238.

[59] W. Chien, J. de Jesus, B. Taylor, V. Dods, L. Alekseyev, D. Shoda, and P. B. Shieh, "The last mile: Dscsa solution through blockchain technology: drug tracking, tracing,

and verification at the last mile of the pharmaceutical supply chain with bruinchain," *Blockchain in Healthcare Today*, vol. 3, pp. 10–30953, 2020.

[60] H. L. Pham, T. H. Tran, and Y. Nakashima, "A secure remote healthcare system for hospital using blockchain smart contract," in *2018 IEEE globecom workshops (GC Wkshps)*, pp. 1–6, IEEE, 2018. https://doi.org/10.1109/GLOCOMW.2018.8644164.

[61] L. Zhou, L. Wang, and Y. Sun, "Mistore: a blockchain-based medical insurance storage system," *Journal of medical systems*, vol. 42, no. 8, p. 149, 2018. https://doi.org/10.1007/s10916-018-0996-4.

[62] Y.-L. Lee, H.-A. Lee, C.-Y. Hsu, H.-H. Kung, and H.-W. Chiu, "Semres-a triple security protected blockchain based medical record exchange structure," *Computer Methods and Programs in Biomedicine*, vol. 215, p. 106595, 2022. https://doi.org/10.1016/j.cmpb.2021.106595.

[63] B. Zaabar, O. Cheikhrouhou, F. Jamil, M. Ammi, and M. Abid, "Healthblock: A secure blockchain-based healthcare data management system," *Computer Networks*, vol. 200, p. 108500, 2021. https://doi.org/10.1016/j.comnet.2021.108500.

[64] K. Azbeg, O. Ouchetto, and S. J. Andaloussi, "Blockmedcare: A healthcare system based on iot, blockchain and ipfs for data management security," *Egyptian informatics journal*, vol. 23, no. 2, pp. 329–343, 2022. https://doi.org/10.1016/j.eij.2022.02.004.

[65] S. M. Pournaghi, M. Bayat, and Y. Farjami, "Medsba: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4613–4641, 2020. https://doi.org/10.1007/s12652-020-01710-y.

[66] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24*, pp. 457–473, Springer, 2005. https://doi.org/10.1007/11426639_27.

[67] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy (SP'07)*, pp. 321–334, IEEE, 2007. https://doi.org/10.1109/SP.2007.11.

[68] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *International workshop on public key cryptography*, pp. 53–70, Springer, 2011. https://doi.org/10.1007/978-3-642-19379-8_4.

[69] D. Riepel and H. Wee, "Fabeo: Fast attribute-based encryption with optimal security," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2491–2504, 2022. https://doi.org/10.1145/3548606.3560699.

[70] T. Feng, F. Kong, C. Liu, and Y. Lu, "Multi-authorization attribute-based verifiable encryption scheme based on blockchain," *Mobile Networks and Applications*, vol. 28, no. 5, pp. 1617–1624, 2023. https://doi.org/10.1007/s11036-023-02099-6.

[71] M. S. Arbabi, C. Lal, N. R. Veeraragavan, D. Marijan, J. F. Nygård, and R. Vitenberg, "A survey on blockchain for healthcare: Challenges, benefits, and future directions," *IEEE communications surveys & tutorials*, vol. 25, no. 1, pp. 386–424, 2022. https://doi.org/10.1109/COMST.2022.3224644.

[72] R. W. Ahmad, K. Salah, R. Jayaraman, I. Yaqoob, S. Ellahham, and M. Omar, "The role of blockchain technology in telehealth and telemedicine," *International journal of medical informatics*, vol. 148, p. 104399, 2021. https://doi.org/10.1016/j.ijmedinf.2021.104399.

[73] T. Kumar, V. Ramani, I. Ahmad, A. Braeken, E. Harjula, and M. Ylianttila, "Blockchain utilization in healthcare: Key requirements and challenges," in *2018 IEEE 20th International conference on e-health networking, applications and services (Healthcom)*, pp. 1–7, IEEE, 2018. https://doi.org/10.1109/HealthCom.2018.8531136.

[74] R. Akkaoui, "Blockchain for the management of internet of things devices in the medical industry," *IEEE Transactions on Engineering Management*, vol. 70, no. 8, pp. 2707–2718, 2021. https://doi.org/10.1109/TEM.2021.3097117.

[75] S. Biswas, K. Sharif, F. Li, Z. Latif, S. S. Kanhere, and S. P. Mohanty, "Interoperability and synchronization management of blockchain-based decentralized e-health systems," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1363–1376, 2020. https://doi.org/10.1109/TEM.2020.2989779.

[76] A. Roehrs, C. A. Da Costa, and R. da Rosa Righi, "Omniphr: A distributed architecture model to integrate personal health records," *Journal of biomedical informatics*, vol. 71, pp. 70–81, 2017. https://doi.org/10.1016/j.jbi.2017.05.012.

[77] G. Zhang, Z. Yang, and W. Liu, "Blockchain-based privacy preserving e-health system for healthcare data in cloud," *Computer Networks*, vol. 203, p. 108586, 2022. https://doi.org/10.1016/j.comnet.2021.108586.

[78] O. Cheikhrouhou, K. Mershad, F. Jamil, R. Mahmud, A. Koubaa, and S. R. Moosavi, "A lightweight blockchain and fog-enabled secure remote patient monitoring system," *Internet of Things*, vol. 22, p. 100691, 2023. https://doi.org/10.1016/j.iot.2023.100691.

[79] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of medical systems*, vol. 42, pp. 1–7, 2018. https://doi.org/10.1007/s10916-018-0982-x.

[80] H. S. Z. Kazmi, F. Nazeer, S. Mubarak, S. Hameed, A. Basharat, and N. Javaid, "Trusted remote patient monitoring using blockchain-based smart contracts," in *Advances on Broad-Band Wireless Computing, Communication and Applications: Proceedings of the 14th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2019) 14*, pp. 765–776, Springer, 2020. https://doi.org/10.1007/978-3-030-33506-9_70.

[81] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "Fhirchain: applying blockchain to securely and scalably share clinical data," *Computational and structural biotechnology journal*, vol. 16, pp. 267–278, 2018. https://doi.org/10.1016/j.csbj.2018.07.004.

[82] R. Akkaoui, X. Hei, and W. Cheng, "Edgemedichain: A hybrid edge blockchain-based framework for health data exchange," *IEEE access*, vol. 8, pp. 113467–113486, 2020. https://doi.org/10.1109/ACCESS.2020.3003575.

[83] Y. Zhuang, L. R. Sheets, Y.-W. Chen, Z.-Y. Shae, J. J. Tsai, and C.-R. Shyu, "A patient-centric health information exchange framework using blockchain technology," *IEEE journal of biomedical and health informatics*, vol. 24, no. 8, pp. 2169–2176, 2020. https://doi.org/10.1109/JBHI.2020.2993072.

[84] X. Liu, A. V. Barenji, Z. Li, B. Montreuil, and G. Q. Huang, "Blockchain-based smart tracking and tracing platform for drug supply chain," *Computers & Industrial Engineering*, vol. 161, p. 107669, 2021. https://doi.org/10.1016/j.cie.2021.107669.

[85] I. A. Omar, R. Jayaraman, K. Salah, I. Yaqoob, and S. Ellahham, "Applications of blockchain technology in clinical trials: review and open challenges," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 3001–3015, 2021. https://doi.org/10.1007/s13369-020-04989-3.

[86] G. Albanese, J.-P. Calbimonte, M. Schumacher, and D. Calvaresi, "Dynamic consent management for clinical trials via private blockchain technology," *Journal of ambient intelligence and humanized computing*, vol. 11, no. 11, pp. 4909–4926, 2020. https://doi.org/10.1007/s12652-020-01761-1.

[87] Y. Zhuang, L. Zhang, X. Gao, Z.-Y. Shae, J. J. Tsai, P. Li, and C.-R. Shyu, "Re-engineering a clinical trial management system using blockchain technology: system design, development, and case studies," *Journal of Medical Internet Research*, vol. 24, no. 6, p. e36774, 2022. https://doi.org/10.2196/36774.

[88] Z. Wenhua, F. Qamar, T.-A. N. Abdali, R. Hassan, S. T. A. Jafri, and Q. N. Nguyen, "Blockchain technology: security issues, healthcare applications, challenges and future trends," *Electronics*, vol. 12, no. 3, p. 546, 2023. https://doi.org/10.3390/electronics12030546.

[89] S. Chenthara, K. Ahmed, H. Wang, F. Whittaker, and Z. Chen, "Healthchain: A novel framework on privacy preservation of electronic health records using blockchain technology," *Plos one*, vol. 15, no. 12, p. e0243043, 2020. https://doi.org/10.1371/journal.pone.0243043.

[90] K. Miyachi and T. K. Mackey, "hocbs: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design," *Information processing & management*, vol. 58, no. 3, p. 102535, 2021. https://doi.org/10.1016/j.ipm.2021.102535.

[91] J. Jayabalan and N. Jeyanthi, "Scalable blockchain model using off-chain ipfs storage for healthcare data security and privacy," *Journal of Parallel and distributed computing*, vol. 164, pp. 152–167, 2022. https://doi.org/10.1016/j.jpdc.2022.03.009.

[92] K. M. Hossein, M. E. Esmaeili, T. Dargahi, A. Khonsari, and M. Conti, "Bchealth: A novel blockchain-based privacy-preserving architecture for iot healthcare applications," *Computer Communications*, vol. 180, pp. 31–47, 2021. https://doi.org/10.1016/j.comcom.2021.08.011.

[93] E. R. D. Villarreal, J. García-Alonso, E. Moguel, and J. A. H. Alegría, "Blockchain for healthcare management systems: A survey on interoperability and security," *IEEE Access*, vol. 11, pp. 5629–5652, 2023. https://doi.org/10.1109/ACCESS.2023.3236505.

[94] L. Soltanisehat, R. Alizadeh, H. Hao, and K.-K. R. Choo, "Technical, temporal, and spatial research challenges and opportunities in blockchain-based healthcare: A systematic literature review," *IEEE Transactions on Engineering Management*, vol. 70, no. 1, pp. 353–368, 2020. https://doi.org/10.1109/TEM.2020.3013507.

[95] K. Yaeger, M. Martini, J. Rasouli, and A. Costa, "Emerging blockchain technology solutions for modern healthcare infrastructure," *Journal of Scientific Innovation in Medicine*, vol. 2, no. 1, 2019.

[96] M. Attaran, "Blockchain technology in healthcare: Challenges and opportunities," *International Journal of Healthcare Management*, vol. 15, no. 1, pp. 70–83, 2022. https://doi.org/10.1080/20479700.2020.1843887.

[97] I. Abu-Elezz, A. Hassan, A. Nazeemudeen, M. Househ, and A. Abd-Alrazaq, "The benefits and threats of blockchain technology in healthcare: A scoping review," *International Journal of Medical Informatics*, vol. 142, p. 104246, 2020. https://doi.org/10.1016/j.ijmedinf.2020.104246.

[98] S. Rahmadika and K.-H. Rhee, "Toward privacy-preserving shared storage in untrusted blockchain p2p networks," *Wireless communications and mobile computing*, vol. 2019, no. 1, p. 6219868, 2019. https://doi.org/10.1155/2019/6219868.

[99] N. Islam, Y. Faheem, I. U. Din, M. Talha, M. Guizani, and M. Khalil, "A blockchain-based fog computing framework for activity recognition as an application to e-

healthcare services," *Future Generation Computer Systems*, vol. 100, pp. 569–578, 2019. https://doi.org/10.1016/j.future.2019.05.059.

[100] B. Shen, J. Guo, and Y. Yang, "Medchain: Efficient healthcare data sharing via blockchain," *Applied sciences*, vol. 9, no. 6, p. 1207, 2019. https://doi.org/10.3390/app9061207.

[101] S. Seebacher and R. Schüritz, "Blockchain technology as an enabler of service systems: A structured literature review," in *Exploring Services Science: 8th International Conference, IESS 2017, Rome, Italy, May 24-26, 2017, Proceedings 8*, pp. 12–23, Springer, 2017. https://doi.org/10.1007/978-3-319-56925-3_2.

[102] S. S. Sarmah, "Understanding blockchain technology," *Computer Science and Engineering*, vol. 8, no. 2, pp. 23–29, 2018. DOI:10.5923/j.computer.20180802.02.

[103] N. Szabo, "Formalizing and securing relationships on public networks," *First monday*, 1997. https://doi.org/10.5210/fm.v2i9.548.

[104] K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi, "Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab," in *Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers 20*, pp. 79–94, Springer, 2016. https://doi.org/10.1007/978-3-662-53357-4_6.

[105] M. Szydlo, "Merkle tree traversal in log space and time," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 541–554, Springer, 2004. https://doi.org/10.1007/978-3-540-24676-3_32.

[106] C. Ye, G. Li, H. Cai, Y. Gu, and A. Fukuda, "Analysis of security in blockchain: Case study in 51%-attack detecting," in *2018 5th International conference on dependable systems and their applications (DSA)*, pp. 15–24, IEEE, 2018. 10.1109/DSA.2018.00015.

[107] X. Yang, Y. Chen, and X. Chen, "Effective scheme against 51% attack on proof-of-work blockchain with history weighted information," in *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 261–265, IEEE, 2019. https://doi.org/10.1109/Blockchain.2019.00041.

[108] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 2567–2572, IEEE, 2017. https://doi.org/10.1109/SMC.2017.8123011.

[109] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004. 10.1109/TAC.2004.834113.

[110] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007. 10.1109/JPROC.2006.887293.

[111] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities," *IEEE access*, vol. 7, pp. 85727–85745, 2019. 10.1109/ACCESS.2019.2925010.

[112] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in *Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99) September 20–21, 1999, Leuven, Belgium*, pp. 258–272, Springer, 1999. https://doi.org/10.1007/978-0-387-35568-9_18.

[113] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer pbft consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2020. https://doi.org/10.1109/TPDS.2020.3042392.

[114] T. A. Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem, and T. Alghamdi, "A comparative analysis of blockchain architecture and its applications: Problems and recommendations," *IEEE access*, vol. 7, pp. 176838–176869, 2019. https://doi.org/10.1109/ACCESS.2019.2957660.

[115] O. Vashchuk and R. Shuwar, "Pros and cons of consensus algorithm proof of stake. difference in the network safety in proof of work and proof of stake," *Electronics and information technologies*, no. 09, 2018. http://dx.doi.org/10.30970/eli.9.106.

[116] B. Sriman, S. Ganesh Kumar, and P. Shamili, "Blockchain technology: Consensus protocol proof of work and proof of stake," in *Intelligent Computing and Applications: Proceedings of ICICA 2019*, pp. 395–406, Springer, 2021. https://doi.org/10.1007/978-981-15-5566-4_34.

[117] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability," in *2019 IEEE international conference on blockchain (Blockchain)*, pp. 536–540, IEEE, 2019. https://doi.org/10.1109/Blockchain.2019.00003.

[118] Caliper-Benchmark, "Caliper." https://github.com/hyperledger/caliper-benchmarks/, 2023.

[119] A. Beimel *et al.*, "Secure schemes for secret sharing and key distribution," *Technion-Israel Institute of technology, Faculty of computer science Haifa ...*, 1996.

[120] Hyperledger, "Hyperledger blockchain performance metrics white paper," *https://www. hyperledger. org/resources/publications/blockchain-performance-metrics*, vol. 31, no. 01, p. 2020, 2018.

[121] N. Singh, "Cpu power and network bandwidth-aware optimal block size computation for blockchain-based applications using meta-heuristic algorithms.," *The Journal of Supercomputing*, vol. 1, no. 16, 2023. 10.1007/s11227-023-05210-6.

[122] R. Arul, R. Alroobaea, U. Tariq, A. H. Almulihi, F. S. Alharithi, and U. Shoaib, "Iot-enabled healthcare systems using blockchain-dependent adaptable services," *Personal and Ubiquitous Computing*, vol. 28, no. 1, pp. 43–57, 2024. https://doi.org/10.1007/s00779-021-01584-7.

[123] S. Xu, J. Zhong, L. Wang, D. He, S. Zhang, and W. Shao, "A privacy-preserving and efficient data sharing scheme with trust authentication based on blockchain for mhealth," *Connection Science*, vol. 35, no. 1, p. 2186316, 2023. https://doi.org/10.1080/09540091.2023.2186316.

[124] Y. Tao, Y. Zhu, C. Ge, L. Zhou, S. Zhou, Y. Zhang, J. Liu, and L. Fang, "Orr-cp-abe: A secure and efficient outsourced attribute-based encryption scheme with decryption results reuse," *Future Generation Computer Systems*, vol. 161, pp. 559–571, 2024. https://doi.org/10.1016/j.future.2024.07.040.

[125] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols: Tool paper," in *International conference on computer aided verification*, pp. 414–418, Springer, 2008. https://doi.org/10.1007/978-3-540-70545-1_38.

[126] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, *et al.*, "The avispa tool for the automated validation of internet security protocols and applications," in *Computer Aided Verification: 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005. Proceedings 17*, pp. 281–285, Springer, 2005. https://doi.org/10.1007/11513988_27.

[127] B. Blanchet, "Automatic verification of security protocols in the symbolic model: The verifier proverif," in *International School on Foundations of Security Analysis and Design*, pp. 54–87, Springer, 2012. https://doi.org/10.1007/978-3-319-10082-1_3.

[128] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The tamarin prover for the symbolic analysis of security protocols," in *Computer Aided Verification: 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25*, pp. 696–701, Springer, 2013. https://doi.org/10.1007/978-3-642-39799-8_48.

[129] Q. Li, Y. Zhang, T. Zhang, H. Huang, Y. He, and J. Xiong, "Htac: Fine-grained policy-hiding and traceable access control in mhealth," *IEEE Access*, vol. 8, pp. 123430–123439, 2020. https://doi.org/10.1109/ACCESS.2020.3004897.

[130] X. Qin, Y. Huang, Z. Yang, and X. Li, "A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing," *Journal of Systems Architecture*, vol. 112, p. 101854, 2021. https://doi.org/10.1016/j.sysarc.2020.101854.

[131] S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro, "A framework and compact constructions for non-monotonic attribute-based encryption," in *Public-Key Cryptography–PKC 2014: 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings 17*, pp. 275–292, Springer, 2014. https://doi.org/10.1007/978-3-642-54631-0_16.

[132] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, pp. 111–128, 2013. https://doi.org/10.1007/s13389-013-0057-3.

[133] A. Thakur, V. Ranga, and R. Agarwal, "Workload dynamics implications in permissioned blockchain scalability and performance," *Cluster Computing*, vol. 27, no. 8, pp. 11569–11593, 2024. https://doi.org/10.1007/s10586-024-04550-z.

[134] T. Suite, "Ganache-truffle suite." https://archive.trufflesuite.com/ganache/, 2024.

[135] O. Lutz, H. Chen, H. Fereidooni, C. Sendner, A. Dmitrienko, A. R. Sadeghi, and F. Koushanfar, "Escort: ethereum smart contracts vulnerability detection using deep neural network and transfer learning," *arXiv preprint arXiv:2103.12607*, 2021. https://doi.org/10.48550/arXiv.2103.12607.

[136] J.-W. Liao, T.-T. Tsai, C.-K. He, and C.-W. Tien, "Soliaudit: Smart contract vulnerability assessment based on machine learning and fuzz testing," in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pp. 458–465, 2019. 10.1109/IOTSMS48152.2019.8939256.

[137] K. L. Narayana and K. Sathiyamurthy, "Automation and smart materials in detecting smart contracts vulnerabilities in blockchain using deep learning," *Materials Today: Proceedings*, vol. 81, pp. 653–659, 2023. https://doi.org/10.1016/j.matpr.2021.04.125.

[138] Y. Wang, X. Zhao, L. He, Z. Zhen, and H. Chen, "Contractgnn: Ethereum smart contract vulnerability detection based on vulnerability sub-graphs and graph neural networks," *IEEE Transactions on Network Science and Engineering*, 2024. https://doi.org/10.1109/TNSE.2024.3470788.

[139] J. Li, G. Lu, Y. Gao, and F. Gao, "A smart contract vulnerability detection method based on multimodal feature fusion and deep learning," *Mathematics*, vol. 11, no. 23, p. 4823, 2023. https://doi.org/10.3390/math11234823.

[140] P. Gong, W. Yang, L. Wang, F. Wei, K. HaiLaTi, and Y. Liao, "Gratdet: Smart contract vulnerability detector based on graph representation and transformer.," *Computers, Materials & Continua*, vol. 76, no. 2, 2023. https://doi.org/10.32604/cmc.2023.038878.

[141] Z. Zhen, X. Zhao, J. Zhang, Y. Wang, and H. Chen, "Da-gnn: A smart contract vulnerability detection method based on dual attention graph neural network," *Computer Networks*, vol. 242, p. 110238, 2024. https://doi.org/10.1016/j.comnet.2024.110238.

[142] V. K. Jain and M. Tripathi, "An integrated deep learning model for ethereum smart contract vulnerability detection," *International Journal of Information Security*, vol. 23, no. 1, pp. 557–575, 2024. https://doi.org/10.1007/s10207-023-00752-5.

[143] L. Zhang, Y. Li, R. Guo, G. Wang, J. Qiu, S. Su, Y. Liu, G. Xu, H. Chen, and Z. Tian, "A novel smart contract reentrancy vulnerability detection model based on bigas," *Journal of Signal Processing Systems*, vol. 96, no. 3, pp. 215–237, 2024. https://doi.org/10.1007/s11265-023-01859-7.

[144] Z. Zhang, Y. Lei, M. Yan, Y. Yu, J. Chen, S. Wang, and X. Mao, "Reentrancy vulnerability detection and localization: A deep learning based two-phase approach," in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pp. 1–13, 2022. https://doi.org/10.1145/3551349.3560428.

[145] M. Rossini, "Slither audited smart contracts dataset," 2022. https://huggingface.co/datasets/mwritescode/slither-audited-smart-contracts.

[146] H. Chu, P. Zhang, H. Dong, Y. Xiao, S. Ji, and W. Li, "A survey on smart contract vulnerabilities: Data sources, detection and repair," *Information and Software Technology*, vol. 159, p. 107221, 2023. https://doi.org/10.1016/j.infsof.2023.107221.

[147] P. Zhang, F. Xiao, and X. Luo, "A framework and dataset for bugs in ethereum smart contracts," in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 139–150, 2020. 10.1109/ICSME46990.2020.00023.

[148] C. Liu, H. Liu, Z. Cao, Z. Chen, B. Chen, and B. Roscoe, "Reguard: finding reentrancy bugs in smart contracts," in *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, pp. 65–68, 2018. https://doi.org/10.1145/3183440.3183495.

[149] J. Feist, G. Grieco, and A. Groce, "Slither: a static analysis framework for smart contracts," in *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in*

Software Engineering for Blockchain (WETSEB), pp. 8–15, IEEE, 2019. https://doi.org/10.1109/WETSEB.2019.00008.

[150] Q. Zhou, K. Zheng, K. Zhang, L. Hou, and X. Wang, "Vulnerability analysis of smart contract for blockchain-based iot applications: A machine learning approach," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24695–24707, 2022. https://doi.org/10.1109/JIOT.2022.3196269.

[151] M. Rossini, M. Zichichi, and S. Ferretti, "On the use of deep neural networks for security vulnerabilities detection in smart contracts," in *2023 IEEE international conference on pervasive computing and communications workshops and other affiliated events (PerCom Workshops)*, pp. 74–79, IEEE, 2023. https://doi.org/10.1109/PerComWorkshops56833.2023.10150302.

[152] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[153] R. Kiani and V. S. Sheng, "Ethereum smart contract vulnerability detection and machine learning-driven solutions: A systematic literature review," *Electronics*, vol. 13, no. 12, p. 2295, 2024. https://doi.org/10.3390/electronics13122295.

[154] B. Lê Hng, T. Lê c, T. oàn Minh, D. Trn Tun, D. Phan Th, and H. Phm Văn, "Contextual language model and transfer learning for reentrancy vulnerability detection in smart contracts," in *Proceedings of the 12th International Symposium on Information and Communication Technology*, pp. 739–745, 2023. https://doi.org/10.1145/3628797.3628945.

[155] M. Rossini, M. Zichichi, and S. Ferretti, "Smart contracts vulnerability classification through deep learning," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pp. 1229–1230, 2022. https://doi.org/10.1145/3560905.3568175.

[156] L. Prifti, B. Cico, and D. Karras, "Smart contract vulnerability detection using deep learning algorithms on evm bytecode," in *2024 13th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–7, IEEE, 2024. https://doi.org/10.1109/MECO62516.2024.10577852.

[157] A. S. Rajasekaran, M. Azees, and F. Al-Turjman, "A comprehensive survey on blockchain technology," *Sustainable Energy Technologies and Assessments*, vol. 52, p. 102039, 2022. https://doi.org/10.1016/j.seta.2022.102039.

[158] A. Thakur, V. Ranga, and R. Agarwal, "Exploring the transformative impact of blockchain technology on healthcare: Security, challenges, benefits, and future outlook," *Transactions on Emerging Telecommunications Technologies*, vol. 36, no. 3, p. e70087, 2025. https://doi.org/10.1002/ett.70087.

[159] T. Fatokun, A. Nag, and S. Sharma, "Towards a blockchain assisted patient owned system for electronic health records," *Electronics*, vol. 10, no. 5, p. 580, 2021. https://doi.org/10.3390/electronics10050580.

[160] P. Zhang, D. C. Schmidt, J. White, and G. Lenz, "Blockchain technology use cases in healthcare," in *Advances in computers*, vol. 111, pp. 1–41, Elsevier, 2018. https://doi.org/10.1016/bs.adcom.2018.03.006.

[161] J. Sengupta, S. Ruj, and S. D. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot," *Journal of network and computer applications*, vol. 149, p. 102481, 2020. https://doi.org/10.1016/j.jnca.2019.102481.

[162] A. Haleem, M. Javaid, R. P. Singh, R. Suman, and S. Rab, "Blockchain technology applications in healthcare: An overview," *International Journal of Intelligent Networks*, vol. 2, pp. 130–139, 2021. https://doi.org/10.1016/j.ijin.2021.09.005.

[163] A. Thakur, D. V. Ranga, and R. Agarwal, "Performance benchmarking and analysis of blockchain platforms," in *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, 2023. https://dx.doi.org/10.2139/ssrn.4385643.

[164] E. A. Shammar, A. T. Zahary, and A. A. Al-Shargabi, "An attribute-based access control model for internet of things using hyperledger fabric blockchain," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 6926408, 2022. https://doi.org/10.1155/2022/6926408.

[165] S. Figueroa, J. Añorga, S. Arrizabalaga, I. Irigoyen, and M. Monterde, "An attribute-based access control using chaincode in rfid systems," in *2019 10th IFIP international conference on new technologies, mobility and security (NTMS)*, pp. 1–5, IEEE, 2019. https://doi.org/10.1109/NTMS.2019.8763824.

[166] A. Pericherla, P. Paul, S. Sural, J. Vaidya, and V. Atluri, "Towards supporting attribute-based access control in hyperledger fabric blockchain," in *IFIP International Conference on ICT Systems Security and Privacy Protection*, pp. 360–376, Springer, 2022. https://doi.org/10.1007/978-3-031-06975-8_21.

[167] I. P. the Distributed Web, "Ipfs powers the distributed web." https://ipfs.tech/, 2023.

[168] A. P. Singh, N. R. Pradhan, A. K. Luhach, S. Agnihotri, N. Z. Jhanjhi, S. Verma, U. Ghosh, D. S. Roy, *et al.*, "A novel patient-centric architectural framework for blockchain-enabled healthcare applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5779–5789, 2020. https://doi.org/10.1109/TII.2020.3037889.

[169] G. Tripathi, M. A. Ahad, and S. Paiva, "S2hs-a blockchain based approach for smart healthcare system," in *Healthcare*, vol. 8, p. 100391, Elsevier, 2020. https://doi.org/10.1016/j.hjdsi.2019.100391.

[170] H. Kasyap and S. Tripathy, "Privacy-preserving decentralized learning framework for healthcare system," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 2s, pp. 1–24, 2021. https://doi.org/10.1145/3426474.

[171] A. Musamih, K. Salah, R. Jayaraman, J. Arshad, M. Debe, Y. Al-Hammadi, and S. Ellahham, "A blockchain-based approach for drug traceability in healthcare supply chain," *IEEE access*, vol. 9, pp. 9728–9743, 2021. https://doi.org/10.1109/ACCESS.2021.3049920.

[172] M. Alnafrani and S. Acharya, "Securerx: A blockchain-based framework for an electronic prescription system with opioids tracking," *Health Policy and Technology*, vol. 10, no. 2, p. 100510, 2021. https://doi.org/10.1016/j.hlpt.2021.100510.

[173] A. A. Abdellatif, L. Samara, A. Mohamed, A. Erbad, C. F. Chiasserini, M. Guizani, M. D. O'Connor, and J. Laughton, "Medge-chain: Leveraging edge computing and blockchain for efficient medical data exchange," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15762–15775, 2021. https://doi.org/10.1109/JIOT.2021.3052910.

[174] S. K. Nanda, S. K. Panda, and M. Dash, "Medical supply chain integrated with blockchain and iot to track the logistics of medical products," *Multimedia Tools and Applications*, vol. 82, no. 21, pp. 32917–32939, 2023. https://doi.org/10.1007/s11042-023-14846-8.

[175] A. Yazdinejad, E. Rabieinejad, T. Hasani, and G. Srivastava, "A bert-based recommender system for secure blockchain-based cyber physical drug supply chain management," *Cluster Computing*, vol. 26, no. 6, pp. 3389–3403, 2023. https://doi.org/10.1007/s10586-023-04088-6.

[176] C. V. B. Murthy and M. L. Shri, "Secure sharing architecture of personal healthcare data using private permissioned blockchain for telemedicine," *IEEE Access*, 2024. https://doi.org/10.1109/ACCESS.2024.3436075.

[177] G. Al-Sumaidaee, R. Alkhudary, Z. Zilic, and A. Swidan, "Performance analysis of a private blockchain network built on hyperledger fabric for healthcare," *Information Processing & Management*, vol. 60, no. 2, p. 103160, 2023. https://doi.org/10.1016/j.ipm.2022.103160.

Proof of Publications

Refereed journals:

Anita Thakur, Virender Ranga and Ritu Agarwal, "Exploring the Transformative Impact of Blockchain Technology on Healthcare: Security, Challenges, Benefits, and Future Outlook," *Transactions on Emerging Telecommunications Technologies* , Wiley, vol. 36, p. e70087, 2025, DOI: https://doi.org/10.1002/ett.70087. **(SCIE, IF=2.5)**.

Transactions on Emerging Telecommunications Technologies

WILEY

SURVEY ARTICLE

Exploring the Transformative Impact of Blockchain Technology on Healthcare: Security, Challenges, Benefits, and Future Outlook

Anita Thakur | Virender Ranga | Ritu Agarwal

Delhi Technological University, Delhi, India

Correspondence: Virender Ranga (virenderranga@dtu.ac.in)

Received: 26 April 2024 | **Revised:** 26 November 2024 | **Accepted:** 14 December 2024

Funding: The authors received no specific funding for this work.

Keywords: blockchain | challenges | healthcare | security | use cases

ABSTRACT

This study aims to methodically explore the diverse applications of blockchain technology (BT) within the healthcare domain. Additionally, it seeks to analyze the inherent challenges of integrating BT into healthcare systems. Furthermore, this study elucidates blockchain's advantageous contributions to the healthcare domain. The suggested research thoroughly reviews the literature from various databases, using predetermined criteria, such as exclusion and inclusion, to identify pertinent studies. It also demonstrates the properties of blockchain and its functionality for the patient, healthcare providers, or overall healthcare infrastructure to assist the healthcare industry in a contemporary direction. The substantial advantages of BT within the medical field are notable. However, it is vital to acknowledge the security vulnerabilities by employing a blockchain-centered strategy to mitigate these challenges, allowing for the creation of a robust and streamlined system and framework. The importance of this study is that as this investigation navigates the convergence of healthcare and BT, it not only delineates the multifaceted applications of BT but also meticulously examines the associated security challenges. The findings of this study chart a course toward a technologically advanced, secure, and efficient healthcare ecosystem, improving patient outcomes and reshaping the future of healthcare delivery worldwide.

1 | Introduction

Healthcare has been an indispensable part of every living being's life. The healthcare sector regularly encounters challenges in providing and retrieving medical records, financial remuneration to healthcare providers, and insurance coverage [1]. The lack of medical records can endanger patients, as doctors may be unaware of crucial details such as current medications, history of laboratory tests, operations, surgeries, and diagnoses. Healthcare record-keeping is subject to tremendous change in developed and developing nations. This vital information is prone to threats like manipulation, deletion, and unauthorized access. To trust another party to keep our data safe could be very dangerous [2].

As per the data furnished by the Protenus breach barometer, 2020 witnessed a breach of over 50 million patient records across 905 distinct incidents, signifying a substantial 44% upsurge in hacking-related breaches. Subsequently, in 2021, approximately 40 million records were compromised, encompassing 758 recorded incidents, with an additional noteworthy 42% increase in breaches attributed to insider threats. In the year 2022, an alarming 50 million-plus records have been compromised in

Anita Thakur, Virender Ranga and Ritu Agarwal, "Workload dynamics implications in permissioned blockchain scalability and performance," *Cluster Computing*, Springer, vol. 27, issue. 8, pp.11569-11593, 2024, DOI: https://doi.org/10.1007/s10586-024-04550-z. **(SCIE, IF=3.6)**

Cluster Computing (2024) 27:11569–11593
https://doi.org/10.1007/s10586-024-04550-z

Workload dynamics implications in permissioned blockchain scalability and performance

Anita Thakur[1] · **Virender Ranga**[1] · **Ritu Agarwal**[1]

Received: 20 December 2023 / Revised: 30 April 2024 / Accepted: 7 May 2024 / Published online: 31 May 2024

Abstract
Blockchain Technology has grown exponentially in recent years due to its decentralized, immutable, transparent data storage, transaction sharing, and processing capabilities. With the emergence of different blockchain platforms, it is important to analyze and evaluate the performance of these platforms in various scenarios. The popularity of the public blockchain, i.e., Bitcoin and Ethereum, has increased manifold. But in distinction to a public blockchain, there is a permissioned and private blockchain that allows restricted involvement of users in the network. To make a well-informed decision regarding the selection of an appropriate platform for utilization, it is crucial to evaluate diverse performance metrics among the numerous available blockchain platforms. In this study, we assessed the performance of the Hyperledger Fabric blockchain (HLF), taking into account various metrics such as resource consumption, throughput, success rate, and latency. We have incorporated parameters such as the ordering service, programming language to write chaincode/smart contracts, number of transactions, transactions per second, and organizations to evaluate the system's performance. Along with our analysis, we also suggested potential areas of research for future development of blockchain technology.

Keywords Hyperledger fabric · Ordering services · Performance metrics · Hyperledger caliper · Chaincode · Benchmarking

1 Introduction

One of the most ground-breaking and ingenious advancements in recent years is blockchain technology. Blockchain is a solid platform that has a wide range of potential uses across many industries. It was first developed as the underpinning technology for the cryptocurrency Bitcoin. With no need for a central authority or middleman, blockchain technology essentially functions as a decentralized, distributed ledger that is well-kept and verified by a network of users. Increased security, transparency, efficiency, and cost-effectiveness are just a few advantages that this technology offers. Blockchain is a topic of tremendous significance for scholars, developers, and business leaders due to its potential to positively change industries, including finance, healthcare, and more. The launch of the Bitcoin network in 2009 helped popularise the technology that has subsequently been employed in many other cryptocurrencies [1].

Through a distributed system, these digital currencies enable the movement of electronic money. Users can use digital signatures to give others ownership of their data, and the transactions are publicly visible on the Bitcoin blockchain. This makes it possible for any user on the network to confirm the validity of transactions independently. The blockchain's decentralized structure makes it possible for the system to work without having a single point of failure that could cause the database as a whole to crash. In a blockchain system, a copy of the ledger is present on every computer in the network at once. This differs from a traditional business where client information is stored on a server farm in a single building. Data loss is a risk even with server farms if proper backups aren't

✉ Anita Thakur
anitathakur_2k21phdit504@dtu.ac.in

Virender Ranga
drvirender.ranga@gmail.com

Ritu Agarwal
ritu.jeea@gmail.com

[1] Department of Information Technology, Delhi Technological University, Delhi 110042, India

Anita Thakur, Virender Ranga and Ritu Agarwal, "Revocable and Privacy-Preserving CP-ABE Scheme for Secure mHealth Data Access in Blockchain," *Concurrency and Computation: Practice and Experience*, Wiley, vol. 37, p. e70064, 2025, DOI: `https://doi.org/10.1002/cpe.70064`. **(SCIE, IF=1.5)**.

Concurrency and Computation: Practice and Experience

WILEY

RESEARCH ARTICLE

Revocable and Privacy-Preserving CP-ABE Scheme for Secure mHealth Data Access in Blockchain

Anita Thakur | Virender Ranga | Ritu Agarwal

Delhi Technological University, Delhi, India

Correspondence: Virender Ranga (virenderranga@dtu.ac.in)

Received: 9 October 2024 | **Revised:** 6 March 2025 | **Accepted:** 14 March 2025

Keywords: attribute-based encryption | blockchain | scyther tool | security

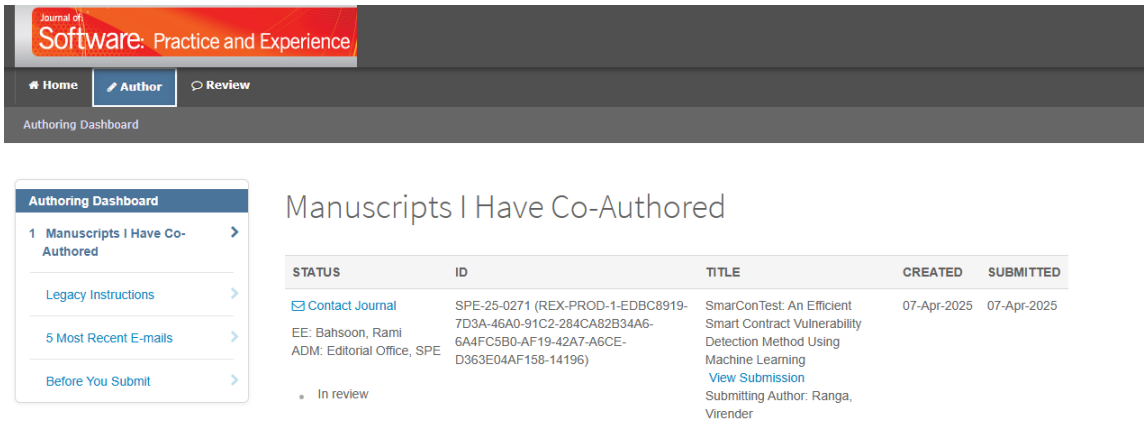
ABSTRACT

Innovations in technology are revolutionizing healthcare, driving a shift toward patient-centric smart healthcare systems. Mobile health (mHealth) leverages innovations in wearable sensors, telecommunications, and IoT to establish a novel healthcare model that prioritizes the patient, enabling real-time monitoring, personalized interventions, and improved access to care, ultimately fostering a proactive approach to health management and enhancing overall patient outcomes. However, safeguarding patient data transparency, security, and privacy within mHealth systems presents significant challenges, particularly concerning personal health records (PHR). Ciphertext-Policy Attribute-Based Encryption (CP-ABE) offers a competent answer to facilitating one-to-many data sharing in healthcare environments. Nevertheless, several issues must be addressed before CP-ABE can be widely deployed. These include the need for timely and effective attribute revocation when user attributes change, resistance to collusion attacks, and ensuring data integrity. This paper proposes a revocable and secure fine-grained access scheme using blockchain and CP-ABE. We compare four prominent state-of-the-art schemes through comprehensive experimentation with our proposed approach. Our results demonstrate the relative performance of our scheme, showing a significant reduction in computational costs. Specifically, the key generation cost is reduced by ≈35% to 67%, and the encryption cost is reduced by ≈26% to 39%. A detailed analysis of communication, computational, and storage overhead reveals that our suggested solution offers a distinct advantage in terms of efficiency. The Scyther tool is employed to verify the security measures and assess the accuracy of proposed methodologies, subsequently conducting experiments to showcase its efficacy.

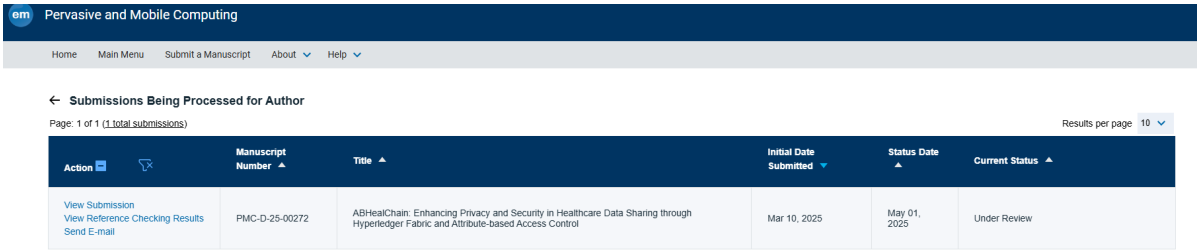
1 | Introduction

The world of healthcare has witnessed a noteworthy evolution with the emergence of mHealth, which builds upon the established foundation of electronic health (eHealth). As a subdomain of eHealth, mHealth capitalizes on the ubiquitous nature of mobile devices to deliver three core functionalities. Firstly, it facilitates the acquisition of health-related data directly from patients. Secondly, the acquired data is securely stored on servers managed by healthcare providers, ensuring its integrity and accessibility. Finally, mHealth empowers disseminating healthcare information to authorized users [1]. By implementing secure, attribute-based, fine-grained access controls, this data distribution can be tailored to specific user roles, such as medical practitioners, researchers, and patients [2]. The healthcare sector generates substantial critical data, presenting valuable research opportunities. However, a primary challenge within the mHealth ecosystem lies in safeguarding the inherent sensitivity of this health information. The potential threat of unauthorized access or attacks necessitates robust security measures to mitigate the risk of data breaches caused by internal and external actors [3]. Furthermore, the online dissemination of sensitive patient

Anita Thakur, Virender Ranga and Ritu Agarwal, "SmarConTest: An Efficient Smart Contract Vulnerability Detection Method Using Machine Learning," —communicated



Anita Thakur, Virender Ranga and Ritu Agarwal, "ABHealChain: Enhancing Privacy and Security in Healthcare Data Sharing through Hyperledger Fabric and Attribute-based Access Control", –communicated



International conferences:

Anita Thakur, Virender Ranga and Ritu Agarwal, "Performance benchmarking and analysis of blockchain platforms", *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, Delhi, pp. 1-7.

Performance Benchmarking and Analysis of Blockchain Platforms

Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022

7 Pages • Posted: 14 Mar 2023

Anita Thakur
Delhi Technological University

Dr. Virender Ranga
Delhi Technological University

Ritu Agarwal
Delhi Technological University

Date Written: March 11, 2023

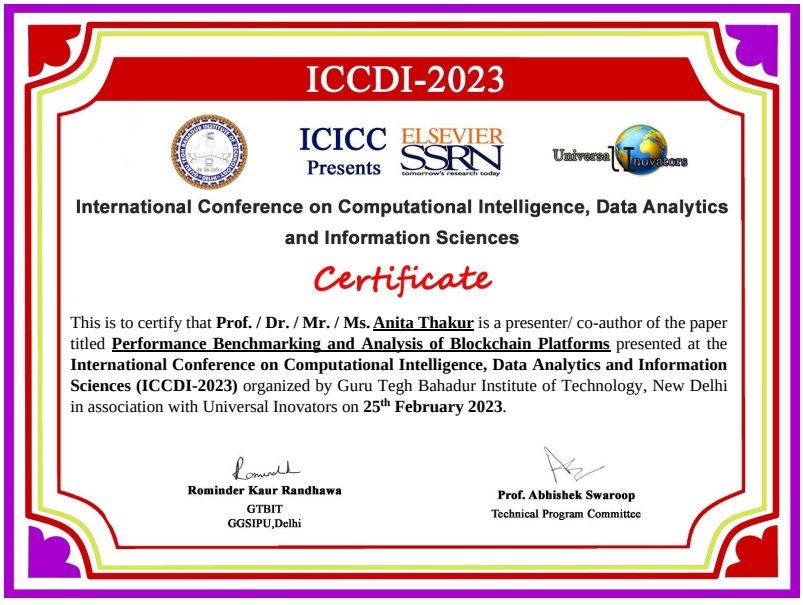
Abstract

The popularity of Blockchain among researchers, developers, and tech-savvy is reaching the sky. Due to the Blockchain's growing attractiveness to the new crowd, anyone familiar with cryptocurrencies must be aware of Blockchain. Blockchain technology provides trust in an untrusted environment of computation and technology. Blockchain has the potential to store, manage and share information in a decentralized manner. This decentralized blockchain property makes this technology robust and immutable to external malicious activities. In this paper, we have conducted a study to measure and assess the performance of various blockchain platforms named Ethereum (private Deployment), Hyperledger Besu Ethereum Client, and Hyperledger Fabric. The performance analysis, such as throughput, resource utilization, and latency, are also calculated. The results show that the Hyperledger Fabric performed very well compared to Ethereum and Hyperledger Besu in all the performance measures.

Keywords: Hyperledger Fabric, Hyperledger Caliper, Ethereum, Hyperledger Besu, Chaincode, Smart Contract, Blockchain

Suggested Citation:

Thakur, Anita and Ranga, Virender and Agarwal, Ritu, Performance Benchmarking and Analysis of Blockchain Platforms (March 11, 2023). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022, Available at SSRN: https://ssrn.com/abstract=4385643 or http://dx.doi.org/10.2139/ssrn.4385643



ICCDI-2023

ICICC Presents ELSEVIER SSRN tomorrow's research today Universal Inovators

International Conference on Computational Intelligence, Data Analytics and Information Sciences

Certificate

This is to certify that **Prof. / Dr. / Mr. / Ms. Anita Thakur** is a presenter/ co-author of the paper titled **Performance Benchmarking and Analysis of Blockchain Platforms** presented at the **International Conference on Computational Intelligence, Data Analytics and Information Sciences (ICCDI-2023)** organized by Guru Tegh Bahadur Institute of Technology, New Delhi in association with Universal Inovators on **25th February 2023**.

Rominder Kaur Randhawa
GTBIT
GGSIPU,Delhi

Prof. Abhishek Swaroop
Technical Program Committee

Anita Thakur, Virender Ranga and Ritu Agarwal, "Attribute-based encryption scheme for secure and efficient access in blockchain", *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, Bangalore, pp. 653-658.

Conferences > 2024 IEEE International Confe...

Attribute-Based Encryption Scheme for Secure and Efficient Access in Blockchain

Publisher: IEEE | Cite This | PDF

Anita Thakur ; Virender Ranga ; Ritu Agarwal All Authors

2 Cites in Papers | 117 Full Text Views

Abstract

Document Sections

I. Introduction

II. Related work

III. Preliminaries

IV. System Model

V. Simulation

Show Full Outline

Authors

Figures

References

Citations

Keywords

Metrics

Abstract:
Due to the large volume of sharing and storage of sensitive information on centralized or distributed storage, there is a growing necessity for employing encryption techniques that provide promising security and access control to the sensitive data on these platforms. Access control is a fundamental aspect of information security, encompassing the mechanisms and policies governing who can access, modify, or use the data within a system. Attribute-based encryption (ABE) is a cryptographic technique that offers a fine-grained access control mechanism by associating attributes with encrypted data and secret keys. This paper suggests the proposed attribute-based encryption as a means to establish secure access control in Blockchain, thereby attaining the secrecy and freshness aspects of cryptographic protocols. The proposed scheme is simulated on SPAN AVISPA. The Security Protocol Animator SPAN functions by simulating an active intruder, enabling interactive identification and construction of potential attacks against various protocols.

Published in: 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)

Date of Conference: 16-17 February 2024

DOI: 10.1109/ICWITE59797.2024.10502721

Date Added to IEEE *Xplore*: 23 April 2024

Publisher: IEEE

ISBN Information:

Conference Location: Bangalore, India

SECTION I.
Introduction

IEEE BANGALORE SECTION | WIE | ICWITE 2024 Engineering Sustainable Futures Responsibly

CERTIFICATE
OF RECOGNITION

This certificate is presented to

Anita Thakur

for participating and presenting the paper titled

Attribute-Based Encryption Scheme for Secure and Efficient Access in Blockchain

in the **2nd IEEE International Conference for Women in Innovation, Technology, and Entrepreneurship**, (ICWITE 2024) on 16-17 Feb 2024 organized by **IEEE Women in Engineering Affinity Group Bangalore Section** at Vemana Institute of Technology, Bangalore.

Prof. Ashwini VR
Chair 2024
IEEE WIE AG Bangalore Section

Dr. Vijayalakshmi K
Chair 2023
IEEE WIE AG Bangalore Section

Dr. P Deepa Shenoy
General Chair
IEEE ICWITE 2024

Anita Thakur, Virender Ranga and Ritu Agarwal, "Simulation-based Performance Evaluation of Consensus Algorithms in NS3 for Blockchain Network", *International Conference on Futuristic Technologies (INCOFT2025)*, Pune.

3rd International Conference on Futuristic Technologies (INCOFT-2025)

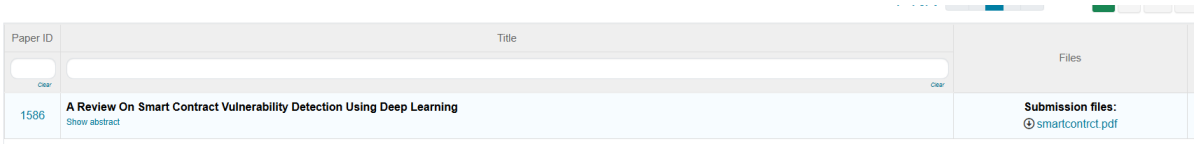
21st – 22nd February 2025

Certificate

This is to certify that Dr./Prof./Mr./Ms. ***Anita Thakur*** *has presented paper entitled* ***Simulation-based Performance Evaluation of Consensus Algorithms in NS3 for Blockchain Network*** *in 3rd International Conference on Futuristic Technologies (INCOFT-2025) during 21st to 22nd February 2025.*

Dr. Chanakya Kumar Jha
General Chair-INCOFT 2025

Anita Thakur, Virender Ranga and Ritu Agarwal, "A Review On Smart Contract Vulnerability Detection Using Deep Learning", —communicated



Paper ID	Title	Files
1586	A Review On Smart Contract Vulnerability Detection Using Deep Learning Show abstract	Submission files: smartcontrct.pdf

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

Plagiarism Verification

Title of the Thesis: **Design and Development of Security Solutions in Blockchain**

Total Pages: **220**

Supervisor(s)

(1) **Dr. Virender Ranga**

(2) **Dr. Ritu Agarwal**

Department: **Information Technology**

This is to report that the above thesis was scanned for similarity detection. The process and outcome are given below:

Software used: **Turnitin** Similarity Index: 4% Word Count: 57460

Date: 27/05/2025

Candidate's Signature **Signature of Supervisor (s)**

Curriculum Vitae

Anita Thakur

Vill. Bhumteer, P.O. Bharai, District and Tehsil Kullu, 175101

anitathakur.25jhn@gmail.com linkedin.com/in/anita-t-928888243/ https://github.com/A4423Y

EDUCATION

Delhi Technological University — **January 2022 – Present**
Doctor of Philosophy — *Delhi, INDIA*

Himachal Pradesh University — **July 2018 – June 2020**
*Master of Technology - **86%*** — *Shimla, Himachal Pradesh*

University Institute of Information Technology (HPU) — **July 2014 – July 2017**
*Bachelor of Technology - **78.44%*** — *Shimla, Himachal Pradesh*

Govt. Polytechnic Sundernagar — **July 2011 – July 2014**
*Diploma - **74.88%*** — *Mandi, Himachal Pradesh*

Govt. Sen. Sec School Bhuthi — **March 2010 – March 2011**
*Matriculation - **89.25%*** — *Kullu, Himachal Pradesh*

COURSEWORK

- Blockchain
- Cryptography
- Computer Networks and Organisation
- Operating Systems
- DBMS
- Computer Architecture

EXAMINATIONS QULIFIED

UGC-NET | Assistant Professor — **2020-June 2021**
- Has qualified UGC-NET for eligibility for Assistant Professor in Computer Science and Application

UGC-NET | Assistant Professor — **2019**
- Has qualified UGC-NET for eligibility for Assistant Professor in Computer Science and Application

PUBLICATIONS

- Thakur, Anita, Dr Virender Ranga, and Ritu Agarwal. "Performance benchmarking and analysis of blockchain platforms." In Proceedings of the International Conference on Innovative Computing & Communication (ICICC). 2022.
- Thakur, Anita, Virender Ranga, and Ritu Agarwal. "Attribute-Based Encryption Scheme for Secure and Efficient Access in Blockchain." In 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE), pp. 653-658. IEEE, 2024.
- Thakur, Anita, Virender Ranga, and Ritu Agarwal. "Simulation-based Performance Evaluation of Consensus Algorithms in NS3 for Blockchain Network." In International Conference on Futuristic Technologies (INCOFT2025), 2024.
- Thakur, Anita, Virender Ranga, and Ritu Agarwal. "Exploring the Transformative Impact of Blockchain Technology on Healthcare: Security, Challenges, Benefits, and Future Outlook." Transactions on Emerging Telecommunications Technologies 36, no. 3 (2025): e70087.
- Thakur, Anita, Virender Ranga, and Ritu Agarwal. "Revocable and Privacy-Preserving CP-ABE Scheme for Secure mHealth Data Access in Blockchain." Concurrency and Computation: Practice and Experience 2025, 37, e70064.
- Gupta, Lakshya, Prateek Jaiswal, Ishita Lather, Ritu Agarwal, and Anita Thakur. "Image Encryption using Chaotic maps: State of the art." In 2023 3rd International Conference on Intelligent Technologies (CONIT), pp. 1-8. IEEE, 2023.
- Thakur, Anita, Virender Ranga, and Ritu Agarwal. "Workload dynamics implications in permissioned blockchain scalability and performance." Cluster Computing (2024): 1-25.
- Thakur, Anita, A.J. Singh. "E-Voting System using Blockchain Technology for Trusted Voting in Democracy"

TECHNICAL SKILLS

Languages: Python, C, HTML, Golang, Solidity
Developer Tools: VS Code, NS3, Docker, RemixIDE, Metamask
Technologies/Frameworks: Linux, GitHub, Hyperledger Fabric, Hyperledger Caliper, Hyperledger Explorer, Ethereum

EXTRACURRICULAR/ACHIEVEMENT/ LEADERSHIP

Certificate of Merit **June 2020**
First position in M.tech *Himachal Pradesh University*

Referral

Dr. Virender Ranga **email: virenderranga@dtu.ac.in**
Associate Professor *Delhi Technological University*

Dr. Ritu Agarwal **email: rituagarwal@dtu.ac.in**
Associate Professor *Delhi Technological University*