Master of Science in Computer Science and Engineering



Storage Efficient Blockchain based Patient Health Record system

by

Sanjida Sharmin

ID: 13MCSE008P

This thesis is submitted in partial fulfillment of the requirement for the degree of
Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chattogram-4349, Bangladesh.
September, 2025

CERTIFICATION

The thesis titled **"Storage Efficient Blockchain based Patient Health Record system"** submitted by **Sanjida Sharmin**, Roll No. **13MCSE008P**, Session **2013-2014** has been accepted as satisfactory in partial fulfillment of the requirement for the degree of **Master of Science in Computer Science and Engineering**

BOARD OF EXAMINERS

1. _____

Dr. Mohammad Shamsul Arefin — Chairman (Supervisor)
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

2. _____

Dr. Pranab Kumar Dhar — Member (Ex-officio)
Professor and Head
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

3. _____

Dr. Asaduzzaman — Member(Internal)
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

4. _____

Dr. Md. Obaidur Rahman — Member (External)
Professor
Department of Computer Science and Engineering
Dhaka University of Engineering Technology (DUET), Gazipur

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate

Sanjida Sharmin
ID: 13MCSE008P

Dedicated to,

my beloved parents

for always supporting me with their endless love and sacrifices.

Acknowledgment

I am overwhelmed to get this opportunity to express my gratitude to those whose constant support and encouragement led me to the completion of this thesis. The satisfaction that accompanies the successful completion of this thesis would be incomplete without the mention of the people whose ceaseless cooperation made it possible. First and foremost, I feel immense proud to express my heartfelt respect and deepest sense of gratitude to my supervisor, Prof. Dr. Mohammad Shamsul Arefin for his continuous guidelines, constructive criticism, inspiration, and overall supervision of this research work. I am ever grateful to the honorable board members for their valuable suggestions and helpful feedbacks. I am also incredibly thankful to all the faculty members and the staff of department of CSE, CUET for their support. Many of my colleagues and friends deserve special thanks for their assistance and motivation.

Finally and always, I am grateful to the Almighty Allah for giving me the strength and patience to complete this thesis work.

Abstract

The protection of sensitive medical information has become a critical concern in modern digital healthcare. This paper introduces a Hybrid Architecture that ensures secure and reliable Healthcare data management through the integration of Blockchain technology with off-chain and on-chain mechanisms. Patient records are encrypted using AES-256-GCM, stored in the InterPlanetary File System (IPFS), and verified using Merkle Tree structures, with only the root values anchored on Ethereum smart contracts. This design guarantees Data Security and integrity while achieving significant Gas Optimization by reducing on- chain storage costs. Experimental evaluation demonstrates that the proposed system achieves high Scalability, efficient transaction processing, and strong resistance to tampering, ensuring confiden- tiality and auditability. By combining Blockchain, cryptographic techniques, and distributed storage, the framework addresses pressing challenges of security, privacy, and trust in healthcare ecosystems. The results highlight the potential of Hybrid Archi- tecture models to deliver a cost-effective, privacy-preserving, and scalable solution for next-generation Healthcare Data Security.

Keywords: Blockchain, Healthcare, AES-256-GCM,Merkle Tree, IPFS, Data Se- curity, Scalability, Gas Optimization, Ethereum, Hybrid Architecture

Contents

Acknowledgment i
Abstract ii

1 Introduction **1**
1.1 Introduction 1
1.2 Blockchain for Healthcare Data Management 1
1.3 Motivation 2
1.4 Challenges 3
1.5 Contributions 5
1.6 Objectives with Specific Aims and Outcomes 6
1.7 Organization of the Report 7
1.8 Conclusion 7

2 Literature Review **8**
2.1 Introducton 8
2.2 Blockchain Architectures in Healthcare 8
2.2.1 Centralized vs. Decentralized Healthcare Models 9
2.2.2 Hierarchical Blockchain Models 9
2.2.3 Hybrid Blockchain with Deep Learning 10
2.2.4 Blockchain with IPFS Storage 11
2.2.5 Layer-2 and Sharding Proposals 12
2.2.6 Smart Contract-Based Access Control 13
2.2.7 Blockchain-IoT-AI Convergence 13
2.2.8 Diagnostic Support with AI-Blockchain Integration 14
2.2.9 Consent and Compliance Frameworks 14
2.2.10 Clinical Trial Monitoring 15
2.3 Comparative Analysis of Prior Works 15
2.4 Research Gaps 16
2.4.1 Research Questions 18
2.5 Conclusion 18

3 Methodology **19**
3.1 Introduction 19
3.2 Detailed Explanation 19
3.2.1 Dataset Preprocessing 19
3.2.2 On-Chain Processing 22
3.2.3 Off-Chain Storage and IPFS Upload 24
3.2.4 Verification and Decryption 28

3.3 Conclusion 29

4 Experimental Results and Evaluation **30**
4.1 Introduction 30
4.2 Experimental Setup and Environment 31
4.2.1 Dataset Description 33
4.2.2 Integrity Verification and Merkle Tree Evaluation 35
4.3 Evaluation Metrics 37
4.3.1 Latency modeling 37
4.3.2 Throughput Modeling 38
4.3.3 Gas Cost matrices 38
4.3.4 Gas Cost Efficiency 38
4.3.5 Gas Cost Analysis in Ethereum 40
4.3.6 Latency Analysis 40
4.4 Throughput Performance 41
4.5 Off-Chain Scalability Analysis 41
4.6 Security Assurance 42
4.7 Comparative Analysis with Prior Works 43
4.7.1 Latency Comparison with Existing Methods 45
4.7.2 Throughput Comparison with Prior Methods 46
4.7.3 Gas Cost Comparison with Prior Healthcare Blockchain Frameworks 47
4.8 Conclusion 48

5 Conclusion and Future Work **49**
5.1 Limitation 49
5.2 Future Work 50
5.3 List of Publications 51

List of Figures

1.1 Identified factors causing scalability issue in public blockchains.[1] 3

2.1 Schematic presentation of the proposed access control and outsourcing through Blockchain. 11

3.1 Dataset preprocessing workflow 21
3.2 Workflow on-chain processing. 22
3.3 Off-chain data processing workflow 24
3.4 Verification process of an encrypted file using SHA-256 hash, Merkle proof, and on-chain Merkle root before decryption. 28

4.1 Sample JSON record representing an encrypted patient file before AES-256 encryption and IPFS upload. 34
4.2 Latency distribution for 40 batches of encrypted patient records. 41
4.3 Throughput distribution across 40 batches, expressed in files per second. 42
4.4 Off-chain scalability analysis: estimated file size distribution across 40 batches. 43
4.5 Latency comparison of the proposed hybrid blockchain system with existing methods across increasing transaction volumes. . . . 46
4.6 Throughput comparison of the proposed system with existing methods under varying transaction volumes. 47

List of Tables

2.1 Strengths and limitations of Key Literature on Blockchain in Healthcare 17

4.1 Sample instances from the raw PMC-Patients dataset. 34
4.2 IPFS CIDv1 (base32) values. 35
4.3 Sample stored SHA-256 hashes of encrypted files. 36
4.4 Merkle roots of file batches. 36
4.5 Merkle proofs of patient records. 36
4.6 Comparative analysis of the proposed hybrid blockchain system against prior works 44
4.7 Comparative Analysis of Gas Cost and Architecture: Proposed System against Prior Work 44

Chapter 1

Introduction

1.1 Introduction

The evolution of digital technologies in healthcare has revolutionized the way medical information is generated, stored, and accessed. With the integration of Electronic Health Records (EHRs), cloud-based services, and Internet of Things (IoT)-enabled medical devices, healthcare systems today generate an enormous volume of sensitive patient data. While these innovations have improved the efficiency and accessibility of healthcare services, they have also introduced serious concerns regarding **data security, privacy, interoperability, and reliability**[1]. Conventional centralized healthcare data systems remain highly vulnerable to cyberattacks, unauthorized access, and single points of failure. Once a central database is compromised, millions of patient records can be exposed, leading to severe consequences in terms of patient trust, compliance with regulations, and operational continuity.

In response to these limitations, blockchain technology has emerged as a promising paradigm for healthcare data management. Its inherent properties-immutability, decentralization, transparency, and cryptographic security—offer strong potential to address the shortcomings of centralized infrastructures.

1.2 Blockchain for Healthcare Data Management

Blockchain is a decentralized, distributed ledger system that records transactions in a tamper-evident manner across multiple nodes. Each transaction is cryptographically linked to its predecessor, ensuring immutability and auditability. Once stored, the data cannot be modified or deleted without detection, making blockchain particularly suitable for trust-critical environments like healthcare.

In the context of healthcare, blockchain offers:

- **Tamper-resistance:** Medical records cannot be altered without detection.
- **Auditability:** Every transaction is timestamped and verifiable.
- **Decentralization:** No single entity controls all patient data.
- **Smart Contracts:** Automated enforcement of access policies, data sharing, and compliance.

When integrated with encryption and decentralized storage, blockchain enables secure, patient-centric healthcare ecosystems, empowering patients with more control over their personal records while ensuring trustworthy collaboration among hospitals, laboratories, insurers, and regulators.

1.3 Motivation

Healthcare systems today generate massive volumes of sensitive data, fueled by EHR platforms, IoT-based health monitoring devices, and cloud-based health services. While these innovations improve accessibility and continuity of care, they introduce severe risks including:

- Unauthorized access and data breaches,
- Centralized control leading to single points of failure,
- Limited interoperability across heterogeneous hospital systems,
- Escalating costs of maintaining secure infrastructures.

Blockchain technology has been proposed as a secure alternative due to its decentralization, immutability, and transparency. However, practical deployment in healthcare faces substantial barriers. Figure 1.1 highlights the relative impact of different factors limiting blockchain adoption.

As shown in Figure 1.1, the most pressing concern is transaction throughput, which directly affects system scalability and the ability to process real-time medical data. Consensus model design is the second most significant factor, as inefficient mechanisms lead to delays and energy overhead. In addition, computation energy requirements and efficiency constraints contribute to performance bottlenecks, while issues such as latency, storage, and transaction verification further hinder large-scale adoption.

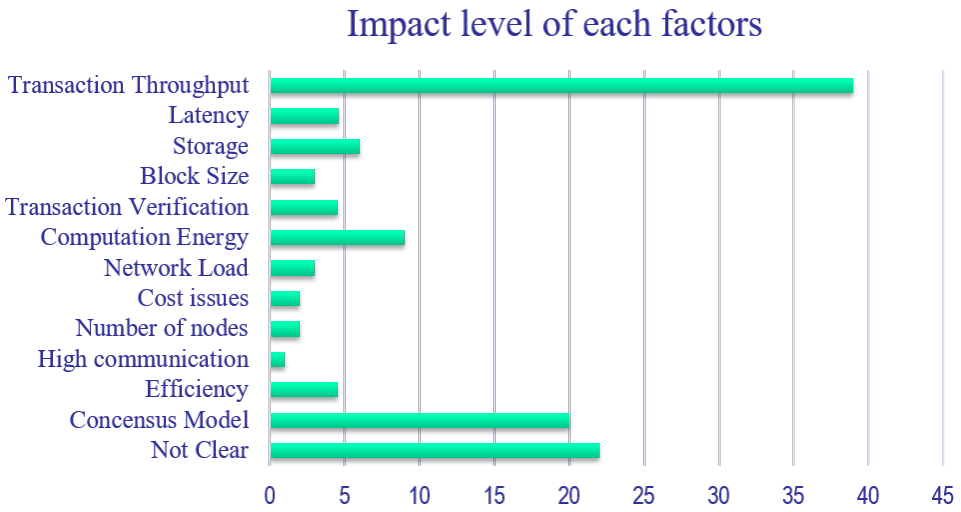


Figure 1.1: Identified factors causing scalability issue in public blockchains.[1]

These findings indicate that conventional public blockchains cannot efficiently handle the high data volume, interoperability needs, and cost requirements of modern healthcare. Therefore, there is a critical need for a hybrid blockchain framework that combines on-chain trust anchoring with off-chain storage, encryption, and batching strategies to achieve both **security** and **scalability**.These challenges highlight the need for a **hybrid blockchain framework** that integrates blockchain with off-chain storage, encryption, and cost-efficient anchoring mechanisms.

1.4 Challenges

The deployment of blockchain in real-world healthcare environments faces several critical challenges that must be addressed to ensure practical adoption and scalability:

- **Scalability:** Most public blockchains, such as Ethereum and Bitcoin, support only a limited number of transactions per second (TPS). In healthcare scenarios, where large volumes of patient records, sensor data, and diagnostic logs must be processed in real time, this restriction can cause significant delays. Peak usage periods—such as during pandemics or mass hospital data transfers—may overwhelm the system, leading to bottlenecks and degraded performance.
- **Gas Costs:** Storing data on-chain is computationally expensive because every node must process and record the information. Healthcare datasets,

which often include detailed medical records and clinical trial data, amplify this cost. As a result, on-chain storage of complete medical files is financially unsustainable, particularly for institutions with limited resources. This necessitates hybrid solutions that anchor minimal metadata or cryptographic proofs on-chain while offloading large files to distributed storage systems.

- **Privacy Concerns:** Even when sensitive patient data is encrypted, transaction metadata (such as timestamps, patient identifiers, or access logs) may unintentionally reveal private information. Because blockchain records are transparent and immutable, adversaries could analyze patterns of access or transaction frequency to infer confidential details about a patient's condition or treatment. Ensuring compliance with privacy regulations like HIPAA or GDPR requires careful design of data masking, access control, and anonymization mechanisms [2, 3].

- **Data Size Limitations:** Blockchains are inherently unsuited for handling large files such as diagnostic imaging (MRI, CT scans), genomic sequences, or high-resolution clinical videos. Attempting to store these directly on-chain would lead to excessive chain bloat, slower consensus, and dramatically higher costs. This limitation makes it essential to design hybrid architectures that leverage off-chain distributed storage networks (e.g., IPFS, cloud systems) while still ensuring verifiable integrity through cryptographic proofs.

- **Interoperability:** Healthcare ecosystems consist of heterogeneous systems—including electronic health records (EHRs), laboratory information systems, and hospital information systems—that use different standards and formats. Achieving seamless blockchain integration across these diverse infrastructures is highly complex. Lack of standardized protocols, varying compliance regulations across regions, and inconsistent adoption of healthcare data formats (like HL7 FHIR) further complicate interoperability. Without overcoming these barriers, blockchain adoption risks creating new data silos instead of enabling secure, universal data exchange.

1.5 Contributions

This thesis addresses the critical challenges of secure, scalable, and cost-effective healthcare data management by proposing a hybrid blockchain framework that integrates on-chain smart contracts with off-chain decentralized storage. The proposed system ensures confidentiality through advanced encryption, reduces gas consumption by leveraging Merkle tree batching, and maintains storage efficiency via distributed IPFS integration. In addition, the framework has been implemented and rigorously evaluated with real-world datasets to validate its practical applicability. The key contributions of this work are summarized as follows:

The key contributions of this thesis are:

1. Hybrid Blockchain Framework: Developed a storage-efficient architecture integrating Ethereum smart contracts with IPFS-based decentralized storage.

2. AES-256-GCM Encryption: Incorporated authenticated encryption for strong confidentiality and integrity of patient records.

3. Merkle Tree Batching: Designed batch-level anchoring of encrypted file hashes, achieving over 99.8% reduction in gas consumption compared to traditional approaches.

4. Storage Efficiency: Demonstrated balanced off-chain storage distribution, with approximately 194–195 MB per batch of 1,000 encrypted files in IPFS, thereby reducing on-chain bloat and ensuring scalability.

5. System Implementation: Implemented the framework using Python, Solidity, IPFS, and Ethereum Sepolia testnet, validated with real datasets.

6. Performance Evaluation: Demonstrated low latency (16–18 seconds), high throughput (up to 1,850 transactions per second), and strong verifiability.

1.6 Objectives with Specific Aims and Outcomes

The specific objective of our work is summarized below:

- Preprocess healthcare datasets to enable secure encryption and blockchain anchoring.
- Apply AES-256-GCM encryption to ensure both confidentiality and authenticity of patient records.
- Employ InterPlanetary File System (IPFS) for decentralized, off-chain storage of large medical files.
- Integrate Merkle tree structures to enable batch verification and cost-efficient anchoring on the blockchain.
- Evaluate system performance in terms of scalability, throughput, latency, and gas efficiency.

Expected Outcomes:

- A storage-efficient framework that minimizes on-chain data bloat by anchoring only Merkle roots while managing large encrypted files through off-chain IPFS storage.
- A cost-efficient blockchain system demonstrating more than 99% reduction in gas cost compared to conventional on-chain storage approaches.
- A privacy-preserving, tamper-evident, and verifiable model for secure healthcare data storage and sharing.
- A practical framework suitable for large-scale, real-world deployment across diverse healthcare institutions.

1.7 Organization of the Report

The thesis is organized as follows:

- **Chapter 2: Literature Review** – Discusses blockchain architectures in healthcare, hybrid frameworks, security-centric models, and regulatory aspects, highlighting research gaps.
- **Chapter 3: Methodology** – Explains the proposed hybrid architecture, covering dataset preprocessing, encryption, IPFS storage, Merkle root construction, and verification.
- **Chapter 4: Results and Analysis** – Presents experimental results including gas efficiency, throughput, latency, scalability, and security assurance.
- **Chapter 5: Conclusion and Future Work** – Summarizes key contributions and outlines directions for future enhancements such as improved privacy and large-scale adoption.

1.8 Conclusion

In conclusion, this research presents a hybrid blockchain-based healthcare data management system that balances security, scalability, and cost-efficiency. By leveraging Ethereum smart contracts and IPFS, the framework ensures data confidentiality, integrity, and verifiability, while significantly reducing on-chain storage costs. The proposed model demonstrates strong potential for real-world adoption and serves as a foundation for future enhancements in decentralized healthcare systems.

Chapter 2

Literature Review

2.1 Introducton

Blockchain technology has emerged as a transformative tool in the healthcare industry, offering capabilities to improve the security, transparency, and efficiency of healthcare data management. As the adoption of Electronic Health Records (EHRs), cloud-based platforms, and Internet of Things (IoT)-enabled medical devices continues to rise, so too do concerns about data breaches, regulatory compliance, and interoperability. Blockchain, with its decentralized, immutable, and cryptographically secure ledger, has shown promise in addressing these challenges. This chapter critically reviews recent advancements in blockchain-based healthcare systems, focusing on architectural models, privacy strategies, security mechanisms, and identified research gaps.

2.2 Blockchain Architectures in Healthcare

Blockchain architectures in healthcare have emerged as a transformative approach to address long-standing issues of data security, interoperability, and patient-centric information management. Unlike traditional centralized health information systems, blockchain enables decentralized storage and access, ensuring that medical records are tamper-proof, transparent, and traceable. Different architectural models—such as public, private, and consortium blockchains—are applied depending on the sensitivity of healthcare data and the level of trust required among stakeholders. For example, permissioned blockchains are often preferred in clinical environments to balance transparency with patient confidentiality, while hybrid approaches integrate both public verification and private data management for scalability[4]. These architectures also support interoperability by

enabling standardized, secure data exchange across hospitals, laboratories, insurers, and patients, which reduces fragmentation and enhances care coordination. However, challenges such as high computational costs, regulatory compliance, and integration with legacy healthcare IT systems remain active areas of research, influencing how blockchain architectures are adapted and optimized for real-world healthcare applications.

2.2.1 Centralized vs. Decentralized Healthcare Models

Traditional healthcare information systems are predominantly built on centralized architectures, where a single authority (such as a hospital, healthcare provider, or third-party cloud platform) controls access to and storage of patient data. While this model offers ease of management and deployment, it suffers from several critical vulnerabilities[5]. The most pressing concern is the single point of failure—if the central server is compromised due to a cyberattack, hardware failure, or internal breach, access to sensitive health data can be disrupted or permanently lost. Moreover, centralized systems often lack comprehensive audit trails, making it difficult to detect unauthorized access or data tampering.

In contrast, decentralized blockchain-based models distribute data across a network of nodes, each maintaining a synchronized copy of the ledger[6]. This architecture ensures fault tolerance, as the failure of a single node does not affect the availability or integrity of the entire system. Data is stored in a tamper-evident manner, protected by cryptographic hash functions and consensus algorithms that validate each transaction before it is added to the chain. This not only enhances data immutability and security but also ensures transparent and auditable access to health records.

Furthermore, blockchain enables patient-centric data ownership, where individuals can grant and revoke access to their health information using cryptographically secured keys and smart contracts. This empowers patients, fosters trust in healthcare IT systems, and supports compliance with privacy regulations such as GDPR and HIPAA.

2.2.2 Hierarchical Blockchain Models

Sadath et al.[7]proposed a hierarchical blockchain architecture designed specifically for healthcare environments, where network scalability and performance are crucial. Their approach divides the blockchain infrastructure into two distinct layers: local chains that operate within individual healthcare institutions (e.g., hospitals, clinics), and a global chain that serves as a higher-level ledger for

synchronizing essential data across organizations. This two-tiered design aims to reduce transaction congestion and optimize resource allocation by enabling local chains to handle frequent, low-level transactions (e.g., patient check-ins, lab result updates) independently, while the global chain records only high-priority or aggregated data.

The hierarchical model improves scalability and throughput by localizing data processing, thereby minimizing the burden on the main blockchain network. It also supports modular deployment, allowing institutions to maintain their own privacy-preserving infrastructure while still contributing to a larger federated system.

However, a significant limitation of Sadath et al.'s framework is its lack of off-chain storage integration. In practice, healthcare data often includes large files such as radiology images, genomic sequences, and medical videos, which are impractical to store directly on-chain due to size constraints and associated gas costs. Without a mechanism for decentralized off-chain storage (e.g., IPFS or cloud-compatible bridges), the model is constrained in handling real-world clinical data at scale. Moreover, the absence of cryptographic proof mechanisms like Merkle trees for verifying off-chain content reduces the system's verifiability and auditability, which are essential in regulated domains like healthcare.

2.2.3 Hybrid Blockchain with Deep Learning

Ali et al. [8]presented an innovative hybrid blockchain framework enhanced by deep learning algorithms to dynamically manage and prioritize healthcare transactions. Their model combines the immutability and decentralized trust of blockchain with the adaptive intelligence of deep learning, creating a system capable of identifying critical medical events and prioritizing them within the blockchain transaction queue. For instance, emergency records or abnormal diagnostic reports could be processed and validated faster than routine check-ups, thereby supporting context-aware healthcare workflows. As shown in Figure 2.1, the access control logic and outsourcing workflow are coordinated via the blockchain layer.

The architecture aims to improve network responsiveness and resource utilization in real-time or near-real-time clinical scenarios. Deep learning models analyze metadata and content from patient records to assess their urgency or relevance, subsequently assigning them weighted priorities. This dynamic prioritization helps overcome the first-in-first-out (FIFO) nature of conventional blockchain systems, which can cause delays in processing urgent medical data.

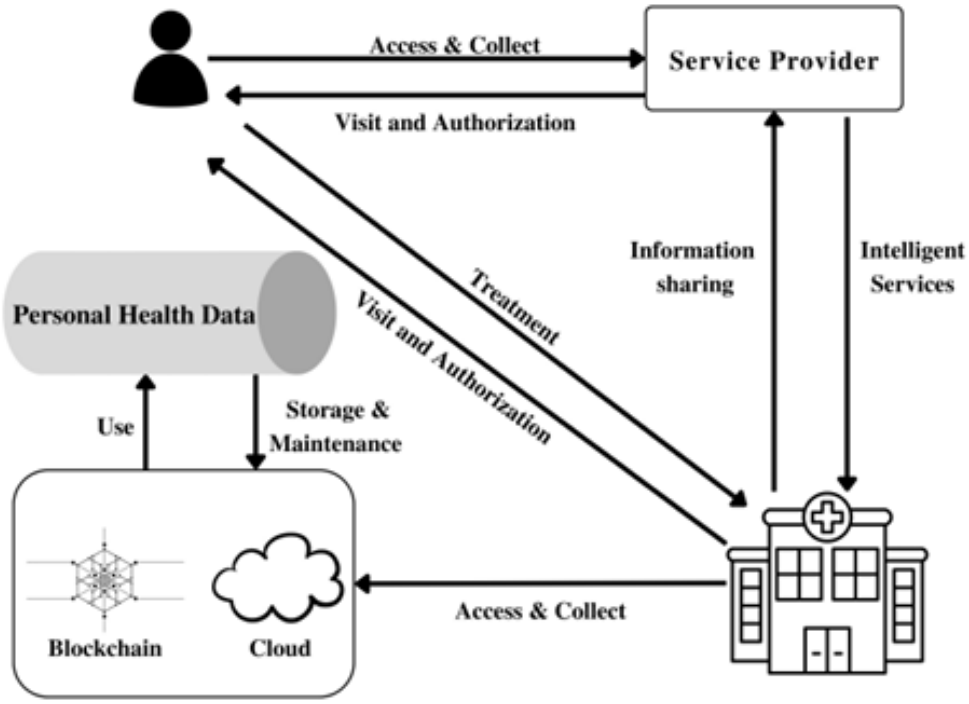


Figure 2.1: Schematic presentation of the proposed access control and outsourcing through Blockchain.

Despite these advantages, the proposed system introduces several drawbacks. The inclusion of deep learning modules significantly increases computational complexity and resource requirements, making deployment challenging in environments with limited infrastructure. Additionally, the decision-making process of deep learning models is often opaque, especially in black-box neural networks, raising concerns over transparency and explainability—critical factors in healthcare, where clinicians must understand and trust system outputs.

Furthermore, the paper does not provide details on how patient data privacy is preserved during deep learning inference, nor does it specify how model updates are handled in a distributed setting. These gaps limit the framework's practical applicability and trustworthiness, particularly in compliance-driven contexts.

2.2.4 Blockchain with IPFS Storage

Jayabalan and Jeyanthi [9] developed a blockchain-based system that integrates the InterPlanetary File System (IPFS) to address on-chain storage limitations commonly encountered in healthcare applications. By offloading large medical files, such as diagnostic reports, EHRs, and imaging data, to IPFS, the model significantly reduces the cost and overhead associated with storing large datasets directly on the blockchain.

The architecture anchors only the file hashes or metadata on-chain, which ensures data traceability while enabling efficient access through content-addressed storage. This approach effectively improves system scalability and enhances patient data availability across distributed environments. It also allows medical institutions to securely share encrypted data over a decentralized network without relying on a single storage provider.

However, one of the major limitations of this model is the absence of batch-wise cryptographic verification mechanisms like Merkle trees. Without such structures, the framework cannot efficiently prove the integrity of grouped data entries, which becomes essential when managing large volumes of medical records. This undermines the overall verifiability and auditability of the stored data, especially when considered for regulatory compliance or forensic traceability. Moreover, the model does not elaborate on fine-grained access control, a critical requirement in privacy-sensitive healthcare systems.

2.2.5 Layer-2 and Sharding Proposals

Mazlan et al. [10] conducted an extensive review of scalability challenges in blockchain systems for healthcare and identified Layer-2 protocols and sharding as potential solutions to improve throughput and latency. Layer-2 techniques such as state channels and sidechains allow certain transactions to be processed off-chain before committing a summarized version to the main chain. This reduces network congestion and lowers transaction fees.

Similarly, sharding involves partitioning the blockchain network into smaller, more manageable pieces (shards), with each shard processing a subset of total transactions in parallel. These strategies are well-established in blockchain literature as effective ways to scale decentralized systems.

Despite their theoretical strengths, the study by Mazlan et al. lacked practical implementation or validation within a healthcare-specific context. The authors discussed architectural principles and conceptual benefits but did not provide deployment frameworks, simulation results, or empirical benchmarks tailored to EHR workloads or real patient datasets. Consequently, the proposed solutions remain conceptual and require further experimentation to establish their feasibility in operational healthcare environments.

2.2.6 Smart Contract-Based Access Control

Shamshad et al.[11] proposed a blockchain-enabled Electronic Health Record (EHR) system that integrates smart contracts for enforcing fine-grained access control policies. In this model, access to patient data is governed by predefined smart contract rules that automatically validate user permissions based on roles, attributes, or temporal constraints. This approach significantly enhances the confidentiality and traceability of data sharing among stakeholders such as clinicians, laboratories, and insurance providers.

The system employs **role-based access control (RBAC)** embedded within smart contracts to ensure that only authorized entities can interact with specific portions of the patient record. Each access request is recorded on-chain, ensuring immutable audit trails that support forensic analysis and regulatory compliance. Moreover, patients retain control over their data by being able to grant or revoke permissions using a user interface linked to the smart contracts.

Despite its strong emphasis on security, the model faces scalability issues, particularly due to per-record transaction processing. Each access event generates an individual blockchain transaction, leading to increased gas costs and latency. This becomes problematic in real-time systems or large-scale hospital networks, where data is accessed frequently. Additionally, the study does not address how the system manages emergency access or revocation propagation across distributed nodes.

2.2.7 Blockchain-IoT-AI Convergence

Mazhar et al.[12] explored a futuristic convergence of blockchain, Internet of Things (IoT), and Artificial Intelligence (AI) to facilitate real-time, personalized healthcare services. Their proposed architecture envisions wearable devices continuously collecting patient vitals (e.g., heart rate, glucose levels), which are then analyzed by AI models to detect anomalies or predict health risks. The processed insights and corresponding events are recorded on a blockchain for secure tracking and automated response.

The system supports a decentralized edge-computing environment, where AI models operate close to data sources (i.e., IoT devices) to minimize latency and preserve patient privacy. Blockchain provides a tamper-proof record of AI-driven decisions, enhancing trust and auditability in automated diagnostics and treatment recommendations. Moreover, smart contracts are used to trigger alerts or initiate interventions (e.g., notifying caregivers) based on AI predictions.

While technically innovative, the framework encounters high implementation

complexity, especially in synchronizing data flows between off-chain IoT systems and the on-chain blockchain ledger. Ensuring data integrity in off-chain environments, especially where network connectivity is intermittent, poses significant challenges. Furthermore, the model lacks a comprehensive strategy for governance and model updates, which are essential in maintaining the accuracy and fairness of deployed AI systems.

2.2.8 Diagnostic Support with AI-Blockchain Integration

Bathula et al.[13] conducted a comprehensive meta-analysis covering more than 400 publications related to the integration of blockchain and artificial intelligence (AI) in healthcare. Their analysis proposed that combining the predictive capabilities of AI with the immutable audit trail of blockchain could significantly enhance diagnostic accuracy and trustworthiness. For example, AI models trained on large-scale medical imaging datasets can identify patterns indicative of diseases, while blockchain can provide secure version control and provenance tracking for diagnostic decisions.

The authors emphasized the conceptual benefits of such integration, particularly for automating radiology diagnostics, pathology screening, and remote patient monitoring. However, the study largely remained at a theoretical level, offering high-level architectural suggestions without detailing how interoperability between AI modules and blockchain layers could be achieved in practice. Crucial implementation aspects such as data annotation standards, off-chain computation handling, and explainability of AI decisions were not explored. As a result, the paper presents a visionary outlook but lacks concrete strategies for deployment in real-world clinical settings.

2.2.9 Consent and Compliance Frameworks

Taherdoost[14] provided a broad review of blockchain applications in managing patient consent and maintaining compliance with regulatory requirements in the healthcare domain. The study acknowledged blockchain's potential to decentralize consent management, enabling patients to control and audit access to their health data using cryptographic keys and smart contracts. This aligns well with evolving privacy norms under the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA).

The review highlighted use cases such as digitally signed consent logs, timestamped access grants, and immutable audit trails, all of which are important for legal and ethical accountability. However, while the study was comprehensive

in scope, it did not present any technical implementation frameworks or validated architectures that align with specific regulatory clauses. Issues like data localization, the right to erasure, and automated consent revocation—critical under GDPR—were not addressed in depth, reducing its applicability in strict compliance-driven environments.

2.2.10 Clinical Trial Monitoring

Kasyapa and Vanmathi [15] examined blockchain's suitability for clinical trial monitoring, particularly in decentralized and remote data collection settings. They argued that blockchain can be used to record trial events such as subject enrollment, treatment administration, and adverse event reports in an immutable and time-ordered fashion. This could significantly improve transparency and reduce fraud in pharmaceutical research.

Their study also emphasized the importance of lightweight consensus mechanisms (e.g., Proof of Authority, Delegated Proof of Stake) and hybrid blockchain architectures to address latency and energy efficiency issues. These are essential considerations in clinical trial settings where real-time updates and mobile data collection are common. However, the analysis remained conceptual, lacking empirical validation or prototype development. As such, while the research provides a useful foundation, further work is needed to assess the performance and scalability of blockchain solutions in active clinical trials.

2.3 Comparative Analysis of Prior Works

The studies reviewed in the previous sections clearly demonstrate the increasing interest in leveraging blockchain technology for healthcare data management. Researchers have explored a wide variety of architectural models, ranging from centralized–decentralized hybrids and hierarchical frameworks to blockchain–IPFS integrations and Layer-2 based scaling solutions. However, these works differ considerably in their scope, with some focusing primarily on access control and authentication, while others emphasize storage efficiency, interoperability, or scalability. They also vary in architectural design, where certain frameworks adopt private or consortium blockchains for greater control, while others rely on public networks to enhance transparency and decentralization. Furthermore, their practical feasibility remains inconsistent, as many proposals remain at a conceptual or simulation stage without large-scale experimental validation, particularly when handling real-world healthcare datasets.

To provide a structured overview of this diverse body of research, Table 2.1 consolidates the key contributions, highlighting the main approaches taken, their respective strengths in addressing specific challenges, and the limitations that hinder their broader applicability. This comparative summary not only underscores the progress made so far but also reveals important research gaps—particularly in areas of cost-efficiency, secure off-chain storage, and verifiable data integrity—that motivate the development of the proposed hybrid blockchain framework presented in this thesis.

2.4 Research Gaps

While the reviewed literature demonstrates the immense potential of blockchain in transforming healthcare data management, several critical gaps persist in the current body of research. These gaps hinder the deployment of fully functional, secure, and scalable blockchain-based healthcare systems in real-world settings. The following key challenges have been identified:

- **Lack of Integrated Frameworks:** Existing studies often focus narrowly on a single aspect (e.g., storage, access control, or transaction prioritization), without offering a comprehensive, end-to-end architecture suitable for real-world healthcare environments.
- **Limited Use of Cryptographic Batching:** Very few frameworks leverage Merkle trees or similar structures for batch-level verifiability, leading to inefficiency in handling large-scale medical records.
- **Inadequate Real-World Validation:** Many models remain conceptual or simulation-based, lacking empirical validation with healthcare datasets or integration into hospital information systems.
- **Cost-Efficiency and Scalability Issues:** On-chain storage is still prohibitively expensive, and most frameworks fail to balance cost optimization with security and scalability in practical deployments.

Table 2.1: Strengths and limitations of Key Literature on Blockchain in Healthcare

Study	Approach	Strengths	Limitations
Sadath et al. [7]	Hierarchical blockchain architecture separating local and global chains	Enhances network scalability and throughput by distributing transactions across layers	No integration with off-chain storage; limited for large data such as medical imaging
Jayabalan et al.[9]	Blockchain integrated with IPFS for decentralized storage	Efficiently reduces on-chain data load and improves scalability for large files	Does not implement Merkle tree-based batch verification, limiting data verifiability
Shamshad et al.[11]	Smart contracts for fine-grained access control in EHR systems	Provides strong data governance and patient-centric access logs	High gas cost and low scalability due to record-level anchoring
Mazhar et al. [12]	Blockchain-IoT-AI hybrid for real-time patient monitoring	Personalized decision-making with edge AI and secure blockchain logging	Implementation complexity and issues with off-chain data synchronization
Taherdoost [14]	Privacy and regulatory compliance review using blockchain	Highlights GDPR/HIPAA alignment and potential in consent logging	No technical implementation; purely conceptual with limited deployment guidance
Bathula et al. [13]	Meta-analysis of AI-blockchain integration in diagnostics	Advocates explainable, auditable diagnostics using immutable AI logs	Lacks implementation roadmap and validation on clinical datasets
Kasyapa and Vanmathi [15]	Blockchain for decentralized clinical trial monitoring	Enables secure trial data collection and transparent auditing	No empirical testing; lacks performance metrics or energy profiling
Ali et al. [8]	Hybrid blockchain with deep learning for transaction prioritization	Improves system responsiveness using AI-based task scheduling	Demands high computation and lacks interpretability of AI model decisions

2.4.1 Research Questions

Based on the identified research gaps, this thesis addresses the following research questions:

1. **RQ1:** How can blockchain-based healthcare systems minimize on-chain storage overhead while ensuring verifiable and tamper-proof access to large-scale medical data?

2. **RQ2:** How can blockchain systems securely store and verify large-scale healthcare records off-chain without compromising privacy or verifiability?

To answer these research questions, we have presented a scalable and privacy-preserving hybrid blockchain framework that integrates AES-256-GCM encryption, IPFS-based decentralized storage, and Merkle tree batch anchoring on Ethereum. The full architecture and implementation details of the proposed method are described in the next chapter.

2.5 Conclusion

This chapter provided a comprehensive review of blockchain applications in healthcare, focusing on architectural designs, security mechanisms, and privacy regulations. While the literature demonstrates significant progress, current solutions are fragmented and often lack scalability or practical deployment strategies. These gaps underscore the need for a robust, hybrid blockchain architecture, as proposed in this study.

Chapter 3

Methodology

3.1 Introduction

This chapter describes the design, processes, and implementation details of the proposed scalable and privacy-preserving hybrid blockchain framework for healthcare data management. The methodology is divided into four key phases: data preprocessing, off-chain encryption and storage, on-chain Merkle root commitment, and data verification with decryption. Each phase has been designed to address core challenges such as data integrity, confidentiality, cost-efficiency, and system scalability in the context of sensitive patient health records.The methodology involves the following key stages, as depicted in Figure 3.1.

3.2 Detailed Explanation

A detailed explanation of each step of the proposed methodology is stated below:

3.2.1 Dataset Preprocessing

The initial phase of the proposed system involves preprocessing a structured healthcare dataset to prepare it for secure encryption, hashing, and metadata extraction. The dataset, typically in CSV format, contains fields such as `patient_uid`, `PMID`, `file_path`, `title`, `patient`, `age`, `gender`, and `relevant_articles`.

Figure 3.1 illustrates the preprocessing pipeline applied to the raw healthcare dataset prior to encryption and blockchain anchoring. The workflow begins with handling missing values by removing records that contain null entries in critical fields such as `patient_uid`, `age`, or `gender`. Following this, duplicate entries are eliminated based on a composite key formed by the `patient_uid` and `PMID`.Once the dataset is cleansed, the `age` field—often stored as nested or

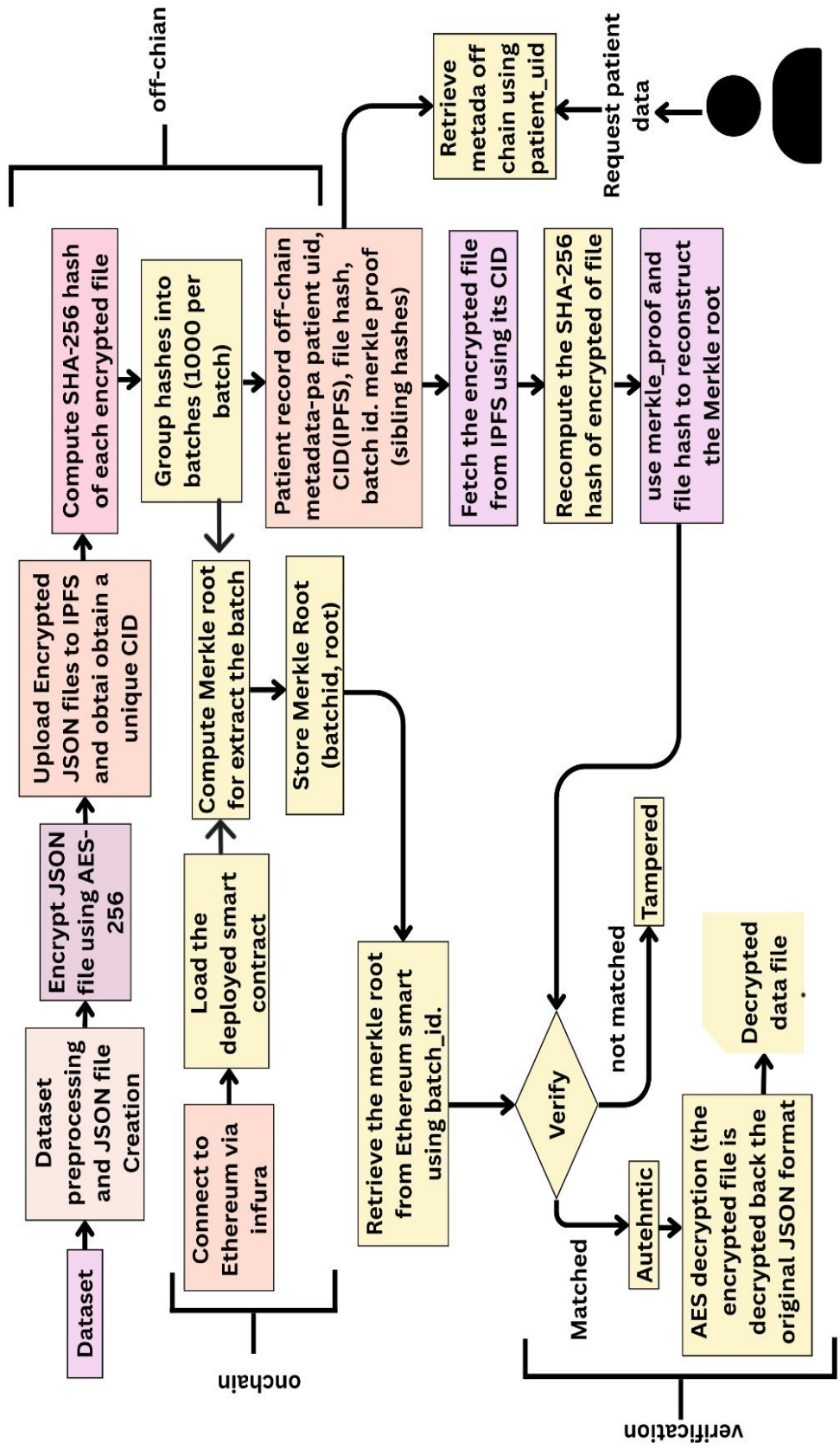


Figure 3.1: System Architecture of the proposed hybrid blockchain system.

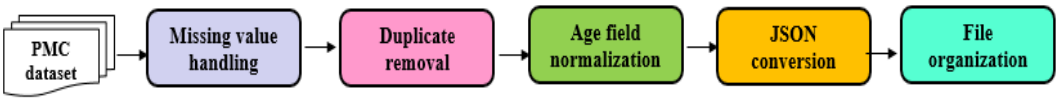


Figure 3.1: Dataset preprocessing workflow

non-uniform formats—is normalized using regular expressions to extract a consistent integer value. Each cleaned record is then converted into a standalone JSON object. These JSON files serve as the input for downstream encryption and off-chain storage. Finally, each JSON file is named using the corresponding `patient_uid` and saved in a structured directory to facilitate organized processing and traceability.This systematic approach ensures that only valid, uniquely identifiable, and standardized records are passed to the encryption and Merkle batching stages of the framework.The detail steps of prepossessing of the dataset is given below:

- **Missing Value Handling:** In the raw dataset, several records contained null or incomplete values in critical fields such as `patient_uid`, `age`, and `gender`. These attributes are essential for uniquely identifying patient records and ensuring reliable downstream processing. To maintain consistency and integrity throughout the encryption and verification pipeline, such incomplete records were discarded. This step prevents errors during JSON serialization and ensures that only valid records progress to subsequent stages.

- **Duplicate Removal:** Duplicate records were detected by constructing a composite key consisting of `patient_uid` and `PMID`. This composite identifier uniquely represents each patient case and its associated publication reference. Removing duplicates eliminated redundancy in the dataset, reduced unnecessary computation in encryption and hashing steps, and ensured that the Merkle tree construction process was not biased by repeated entries.

- **Age Field Normalization:** The `age` attribute was often stored in inconsistent or nested structures (e.g., `[[60.0, "year"]]`) within the dataset. To standardize this field, a regular expression parser was applied to extract the numeric value (e.g., `60`) and convert it into an integer format. This normalization step guaranteed uniform representation of patient ages, facilitating accurate metadata handling and enabling downstream analytics

without ambiguity.

- **JSON Conversion:** Each cleaned record was then serialized into a standalone JSON object, containing both the patient attributes and the relevant metadata. These JSON objects formed the direct input to the AES-256-GCM encryption module. Moreover, the metadata embedded in these objects (e.g., patient identifiers, file paths, and references) was crucial for computing file hashes, building Merkle proofs, and ensuring seamless traceability in the verification process.

- **File Organization:** To enable systematic processing, each JSON object was saved as an individual file and assigned a filename derived from its `patient_uid`. This organization ensured that files were uniquely named, easily retrievable, and traceable throughout the workflow. The

  This preprocessing pipeline ensures the dataset is complete, uniquely identifiable, and structurally standardized before it enters the AES-256 encryption and Merkle tree computation stages.

3.2.2 On-Chain Processing

The on-chain component of the proposed system is responsible for storing cryptographic summaries (Merkle roots) of encrypted patient record batches to ensure tamperproof verification. Figure 3.2 illustrates the on-chain data processing phase in the proposed framework. The process begins with SHA-256 hashing of encrypted patient records, ensuring each file has a unique digital fingerprint. This is followed by the AES-256-GCM encryption stage that protects data confidentiality and integrity through secure encoding and authentication tags.

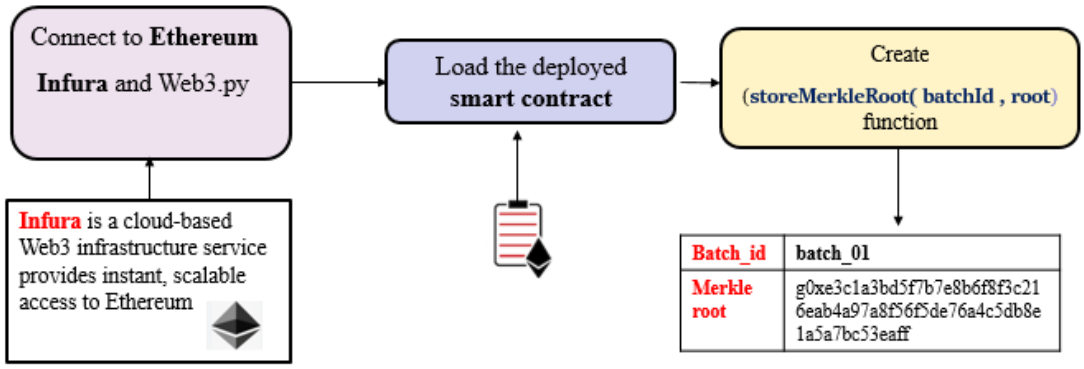


Figure 3.2: Workflow on-chain processing.

Figure 3.2 illustrates the on-chain data processing phase in the proposed framework. The process begins with SHA-256 hashing of encrypted patient records, ensuring each file has a unique digital fingerprint. This is followed by the AES-256-GCM encryption stage that protects data confidentiality and integrity through secure encoding and authentication tags.

Next, the encrypted files are uploaded to the InterPlanetary File System (IPFS), a decentralized storage platform. The resulting Content Identifiers (CIDs) are used to reference the files in a blockchain-compatible manner[16]. These CIDs are hashed and grouped into Merkle trees, and the resulting Merkle roots are committed to the Ethereum blockchain via a smart contract. This anchoring step guarantees tamper-evident verification and provides immutable references to large batches of encrypted healthcare data. The process is carried out using smart contracts deployed on the Ethereum Sepolia testnet and consists of the following steps:

- **Network Connection:** The system establishes a secure connection to the Ethereum Sepolia testnet through **Infura**, which acts as a reliable remote Ethereum node provider. This eliminates the need to run a full Ethereum node locally, thereby reducing overhead and improving accessibility. The `Web3.py` Python library is employed to handle blockchain interactions such as sending transactions, retrieving stored values, and monitoring events emitted by the smart contract. This integration ensures seamless communication between the off-chain Python environment and the on-chain Ethereum infrastructure.
- **Smart Contract Integration:** A lightweight smart contract, implemented in **Solidity**, is deployed on the Ethereum Sepolia testnet to manage Merkle root commitments. The contract provides methods for both storeMerkleRoot(batchId, root) to anchor the Merkle root of a batch on-chain and getMerkleRoot(batchId) to retrieve it for verification. By design, the contract enforces access control so that only the contract owner (deployer) can submit new Merkle roots, preventing unauthorized modifications. All interactions with the contract are logged as immutable blockchain transactions, ensuring transparency and auditability.
- **Merkle Root Storage:** For each batch of 1,000 encrypted patient records, individual SHA-256 hashes are first computed and used as the leaf nodes of a Merkle tree. The Merkle tree construction iteratively

combines these hashes until a single root value is obtained, which compactly represents the integrity of the entire batch. The resulting Merkle root, along with its corresponding batch identifier, is submitted to the blockchain via the smart contract function `storeMerkleRoot(batchId, root)`. Anchoring only the Merkle root instead of every file hash drastically reduces on-chain storage costs while still preserving verifiability. Any stakeholder can later recompute the hash of a downloaded encrypted file, obtain its Merkle proof, and verify consistency with the on-chain Merkle root, ensuring both **immutability** and **decentralized verification**.

3.2.3 Off-Chain Storage and IPFS Upload

The off-chain component handles encryption, decentralized file storage, hash computation, and metadata management.

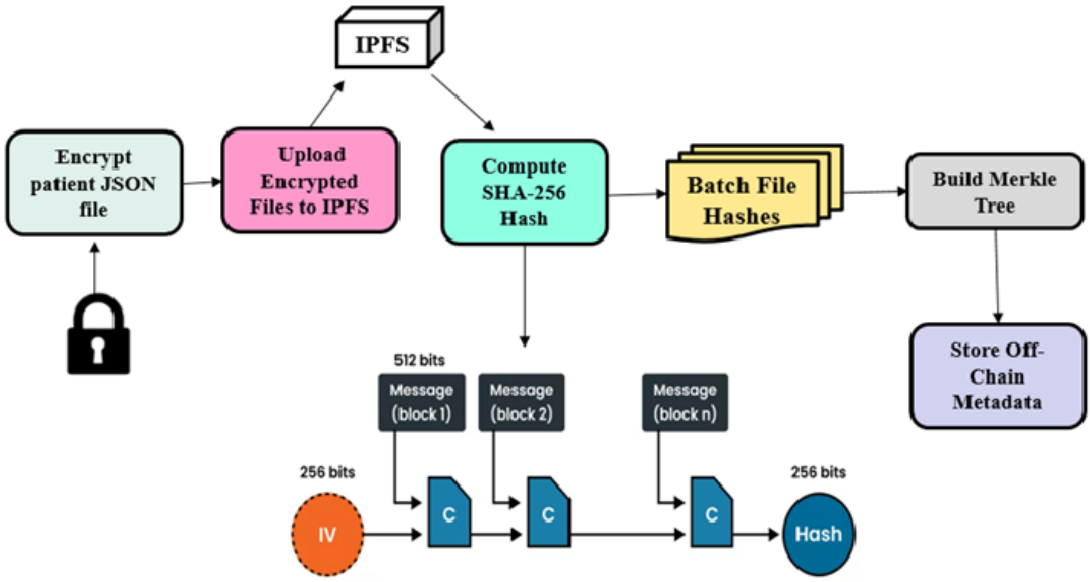


Figure 3.3: Off-chain data processing workflow

Figure 3.3 presents the off-chain data handling process used in the proposed hybrid blockchain framework. The workflow begins with encrypting each patient's JSON-formatted health record using the AES-256-GCM encryption algorithm, ensuring both confidentiality and integrity.This process is organized into six key steps:

– **Encrypt Patient JSON Files** — Each patient's data is serialized into JSON format and encrypted using AES-256-GCM[17, 18].

This encryption ensures both data confidentiality and integrity. The full procedure of encryption is presented in Algorithm 1.The algorithm encrypts a patient record D with AES-256-GCM to produce$E = N \,\|\, C \,\|\, T$, where N is a 12-byte nonce, C is the ciphertext, and T is a 16-byte authentication tag. Inputs are the serialized record D, a 256-bit key K, and a nonce N that must be unique per K; optional associated data (AAD) may be bound to the tag. Steps: initialize the cipher with (K, N), encrypt the plaintext (and AAD) to obtain (C, T), then return E. Security: provides confidentiality and authenticated integrity; any change to N, C, or T is detected at decryption; nonce reuse under the same key is not allowed. Overhead is 28 bytes per record; time complexity is $\mathcal{O}(|D|)$. In our pipeline, the SHA-256 digest is computed over E for Merkle commitments.

– **Upload Encrypted Files to IPFS** — The encrypted files are uploaded to the InterPlanetary File System (IPFS)[19], a distributed storage network. The returned content identifier (CID) for each file is saved for future reference and verification.

Algorithm 1: AES-256-GCM Encryption of Patient Records

Input:
D : Patient JSON record
K : 256-bit AES symmetric key
N : 12-byte random nonce

Output:
E : Encrypted file containing $\{nonce, ciphertext, tag\}$

$plaintext \leftarrow \textsc{Serialize}(D)$
$cipher \leftarrow \textsc{AesGcmInit}(K, N)$
$(ciphertext, tag) \leftarrow cipher.\textsc{encrypt}(plaintext)$
$E \leftarrow N \,\|\, ciphertext \,\|\, tag$
return E

– **Compute SHA-256 Hash** — For each encrypted file, the SHA-256 [20] hash is computed. This generates a fixed-length 256-bit digest that serves as a unique digital fingerprint of the file. Even the smallest modification to the encrypted file produces a completely different hash value, thereby ensuring strong tamper detection and data integrity. These SHA-256 digests act as the *leaf nodes* of the Merkle tree and form the foundation for subsequent verification.

– **Batch File Hashes** — Once the file hashes are generated, they are grouped into batches of 1,000 files each. Every batch is processed inde-

pendently to construct its corresponding Merkle tree [21, 22]. Batching reduces on-chain storage requirements significantly, since only one Merkle root per batch needs to be anchored to the blockchain. This approach enhances scalability and efficiency, while still allowing any single file to be verified against the on-chain root using a Merkle proof.

– **Build Merkle Tree** — Algorithm 2 computes Merkle roots from batches of encrypted patient files using the *SHA-256* hash function. The process begins by hashing each encrypted file f_i to obtain a set of leaf nodes. These leaves are then partitioned into fixed-size batches (e.g., 1,000 per batch).For each batch, the algorithm ensures that the number of leaves is a power of two by duplicating the last element if necessary.It then iteratively combines pairs of hashes at each level: for every two adjacent nodes, their concatenation is hashed to form the parent node. This process continues until only a single node remains, which represents the Merkle root of the batch.The final output is a set of Merkle roots, one per batch, which are subsequently anchored on-chain. These roots serve as compact cryptographic commitments, enabling efficient verification of any individual encrypted file through its Merkle proof, without requiring all records to be stored on the blockchain.

– **Store Off-Chain Metadata** — For each file, corresponding metadata including patient UID, CID, file hash, batch ID, and Merkle proof is compiled and stored in a structured CSV file. This metadata supports efficient file retrieval and integrity verification.

Algorithm 2: Merkle Root Construction from Encrypted Files

Input:

$F = \{f_1, f_2, \ldots, f_n\}$: encrypted files (bytes)

B: batch size (e.g., 1000)

Hash function $H(\cdot)$: SHA-256

Output:

$R = \{r_1, r_2, \ldots, r_k\}$: Merkle roots, one per batch

$L \leftarrow [\,]$ {leaf hashes}
for each file $f_i \in F$ **do**
  $h_i \leftarrow H(f_i)$
  append h_i to L
end for
Partition L into batches $\{L^{(1)}, L^{(2)}, \ldots, L^{(k)}\}$ of size B
$R \leftarrow [\,]$
for each batch $L^{(j)}$ **do**
  while $|L^{(j)}|$ is not a power of 2 **do**
    append last element of $L^{(j)}$ to $L^{(j)}$ {duplicate last leaf}
  end while
  $S \leftarrow L^{(j)}$ {current level}
  while $|S| > 1$ **do**
    $N \leftarrow [\,]$ {next level}
    for $t = 1$ to $|S|$ step 2 **do**
      $p \leftarrow H\big(S[t] \,\|\, S[t+1]\big)$ {left $\|$ right}
      append p to N
    end for
    $S \leftarrow N$
  end while
  $r \leftarrow$ the only element of S {Merkle root for batch j}
  append r to R
end for
return R

3.2.4 Verification and Decryption

To verify the integrity of any encrypted file, its SHA-256 hash is first recomputed. This hash, together with the corresponding Merkle proof, is then used to reconstruct the Merkle root of the batch. The reconstructed root is compared against the on-chain Merkle root stored in the Ethereum smart contract. If both roots match, the file is confirmed to be authentic and untampered. Only after successful verification is the file decrypted using the AES-256 key, ensuring that access is granted exclusively to trusted data. This process preserves both security and reliability of the healthcare records. The overall verification and decryption workflow is illustrated in Figure 3.4, while the detailed decryption procedure is presented in Algorithm 3.

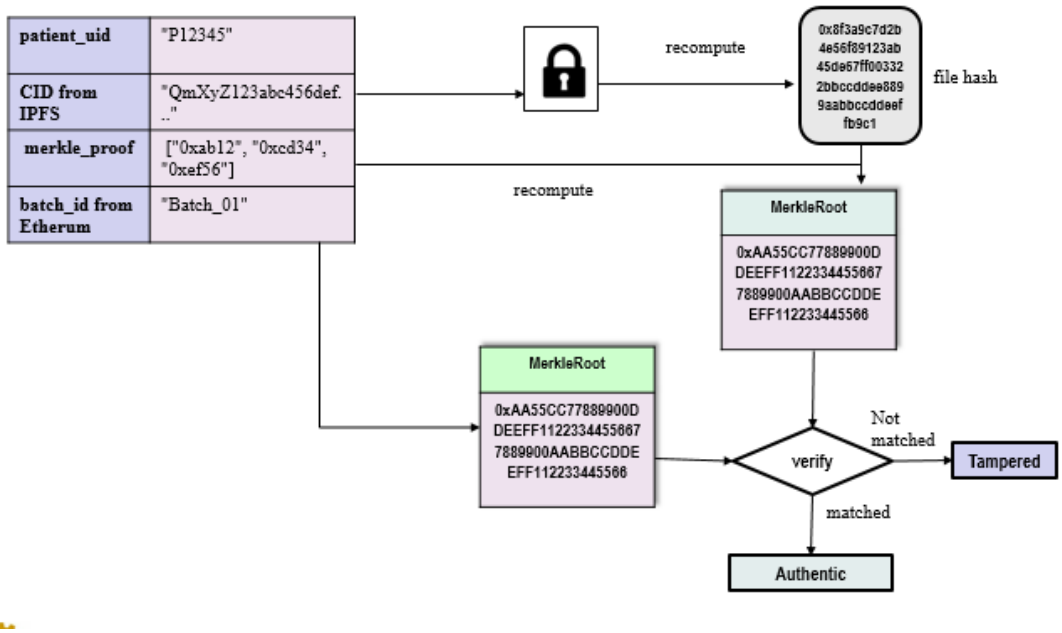


Figure 3.4: Verification process of an encrypted file using SHA-256 hash, Merkle proof, and on-chain Merkle root before decryption.

Algorithm 3: AES-256-GCM Decryption of Patient Records

Input:

E: envelope bytes $E = N \,\|\, C \,\|\, T$

K: 256-bit AES decryption key (32 bytes)

Output:

D: decrypted patient JSON record

$N \leftarrow E[0{:}12]$ {first 12 bytes = nonce}
$T \leftarrow E[-16{:}]$ {last 16 bytes = tag}
$C \leftarrow E[12{:}-16]$ {middle segment = ciphertext}
$aead \leftarrow$ InitAES_GCM(K)
$plaintext \leftarrow aead$.decrypt($N$, C, T) {verifies tag before release}
if decrypt failed **then**

  return AuthError
end if
$D \leftarrow$ ParseJSON(UTF8Decode($plaintext$))
return D

3.3 Conclusion

The methodology outlined in this chapter establishes a robust framework for scalable and secure healthcare data management using blockchain. By combining AES-256-GCM encryption, Merkle tree batching, IPFS-based decentralized storage, and Ethereum smart contracts, the system achieves significant improvements in privacy, gas efficiency, and verifiability. The staged approach from preprocessing to verification ensures integrity and performance across a large volume of healthcare records, making the solution suitable for real-world, privacy-sensitive medical applications.

Chapter 4

Experimental Results and Evaluation

4.1 Introduction

This chapter details the implementation and evaluation of the proposed hybrid blockchain architecture. It first introduces the dataset and preprocessing pipeline, including the conversion of records into standalone JSON objects for encryption and decentralized storage. The experimental environment is then described, highlighting the programming language, Python libraries, and supporting platforms used to integrate cryptographic operations, decentralized storage through IPFS, and blockchain functionality via Ethereum Sepolia testnet.Smart contracts, implemented in Solidity and deployed with Hardhat, were accessed through Infura and Web3.py, enabling secure storage and retrieval of Merkle roots associated with encrypted patient record batches. The hardware configuration used for the experiments is also specified to ensure reproducibility.

Following this, the evaluation methodology is presented, covering cryptographic validation, IPFS-based storage integrity, gas cost analysis, and Merkle tree verification. Performance comparisons with existing approaches are provided, alongside analyses of scalability, transaction latency, and throughput under varying network conditions. Finally, the chapter discusses the stability of the system and presents visualizations of the verification and decryption processes, demonstrating how the proposed framework ensures privacy, security, and efficiency in healthcare data management.

4.2 Experimental Setup and Environment

The experimental setup for evaluating the proposed hybrid blockchain framework involved multiple tools, technologies, and environments. The system was designed to ensure secure encryption, decentralized storage, efficient blockchain anchoring, and real-world performance validation. The key components of the setup are outlined below:

- **Programming Language and Environment:**
  - The entire implementation was developed using **Python 3.x**.
  - All scripts were executed within **Google Colab**, a cloud-based Jupyter notebook platform providing ease of access, resource allocation, and seamless integration with external APIs.

- **Cryptographic Operations:**
  - `cryptography` library was used to perform AES-256-GCM encryption and decryption, ensuring confidentiality and data integrity of patient records.
  - `hashlib` library was employed to compute SHA-256 hashes of the encrypted files. These hashes served as Merkle tree leaves to support batch-level verification.

- **Decentralized Storage (IPFS):**
  - **IPFS Desktop** and `go-ipfs` were used to host the encrypted files in a decentralized peer-to-peer manner.
  - On successful upload, a unique **Content Identifier (CID)** was returned and stored for traceability.

- **Blockchain and Smart Contract Integration:**
  - Smart contracts were developed in Solidity and deployed to the Ethereum Sepolia testnet.
  - `Hardhat` was used as the Solidity development framework for compiling, testing, and deploying contracts.
  - Blockchain transactions were executed using the `Web3.py` library in Python.

- **Ethereum Node Provider (Infura):**
  - **Infura** API gateway was used to connect Google Colab and Python scripts to the Ethereum Sepolia testnet without the need to run a local Ethereum node.
  - It provided a reliable RPC endpoint for broadcasting transactions, querying blockchain state, and retrieving smart contract data.

- **Hardware Configuration:**
  - All experiments were executed on a local machine with the following specifications:
    - Processor: **Intel(R) Core(TM) i7-4510U CPU** @ 2.00 GHz (Turbo up to 2.60 GHz)
    - RAM: **6 GB**
    - Operating System: **Windows 10 Pro (64-bit)**

- **Key Python Libraries:**
  - `cryptography`: Used for AES-256-GCM encryption and decryption
  - `hashlib`: Used for SHA-256 hashing of encrypted files
  - `web3.py`: Enables Python-based interaction with Ethereum smart contracts
  - `ipfshttpclient`: Handles file uploads and retrievals from IPFS
  - `pandas, json, os`: For dataset parsing, file management, and metadata processing

4.2.1 Dataset Description

For the experimental evaluation, this study employed the **PMC-Patients dataset**, which comprises patient summaries extracted from clinical case reports available in the *PubMed Central (PMC)* repository. This dataset was selected due to its richness in clinical narratives and structured metadata, making it suitable for testing blockchain-based health record storage and verification frameworks. The raw dataset contained **40,000 patient records**. Each record included key fields such as:

- **patient_id** – Unique identifier for each entry in the dataset.
- **patient_uid** – An anonymized identifier mapped to each patient.
- **PMID** – The PubMed reference ID linking the record to the original clinical report.
- **file_path** – The storage reference of the extracted case report file.
- **title** – Title of the case study or medical report.
- **patient summary** – A narrative description of the patient case, often including age, condition, and treatment details.
- **age** – Age of the patient, sometimes stored in irregular formats (e.g., `[[60.0, 'year']]`).
- **gender** – Gender of the patient.
- **relevant_articles** – References to related PubMed articles supporting the case report.

Table 4.1 presents representative instances of the raw dataset prior to preprocessing. These records demonstrate the diversity of metadata fields and the semi-structured nature of the data.Since raw clinical data often contains inconsistencies, a data preprocessing pipeline was applied to prepare the dataset for downstream encryption and blockchain anchoring. The preprocessing steps included:

1. **Duplicate Removal** – Eliminating records with identical *patient_uid* and *PMID* pairs.
2. **Handling Missing Values** – Dropping incomplete entries where critical fields (*patient_uid, age, gender*) were missing.
3. **Normalization of Age** – Parsing nested age fields into standardized integer values (e.g., converting `[[60.0, 'year']]` to "60").

"patient_uid":	**"7665777-1",**
"pmid":	33492400,
"file_path":	"comm/PWC007xxxxxxx/PMC7665777.xml",
"title":	"Early Physical Therapist Interventions for Patients with COVID-19 in the Acute Care Hospital: A Case Report Series",
"patient":	"This 60-year-old male was hospitalized due to moderate ARDS from COVID-19 with symptoms of fever, dry cough, and dyspnea.",
"age":	60,
"gender":	"M",
"relevant_articles	": { "32320506": 1, "32239716": 1, "23219649": 1, "30339549": 1, "17470624": 1, "32280973": 1, "34789437": 1, "30427933": 1 }

Figure 4.1: Sample JSON record representing an encrypted patient file before AES-256 encryption and IPFS upload.

4. **Serialization** – Each cleaned record is converted into a standalone JSON object, which is then used for encryption and off-chain storage, as shown in Figure 4.1.
5. **Resulting Dataset Size** – Reduced from 40,000 to 39,936 files after preprocessing.

Table 4.1: Sample instances from the raw PMC-Patients dataset.

id	patient_uid	PMID	file_path	title	patient	age	gender	relevant_articles
0	7665777-1	33492400	comm/PMC…	Early Physic…	This 60-yea…	[[60.0, 'year']]	M	{'32320506': 1, …}
1	7665777-2	33492400	comm/PMC…	Early Physic…	A 39-year-c…	[[39.0, 'year']]	M	{'32320506': 1, …}
2	7665777-3	33492400	comm/PMC…	Early Physic…	One week a…	[[57.0, 'year']]	M	{'32320506': 1, …}
3	7665777-4	33492400	comm/PMC…	Early Physic…	This 69-yea…	[[69.0, 'year']]	M	{'32320506': 1, …}
4	7665777-5	33492400	comm/PMC…	Early Physic…	This 57-yea…	[[57.0, 'year']]	M	{'32320506': 1, …}

4.2.2 Integrity Verification and Merkle Tree Evaluation

In this work, the outputs of the proposed hybrid blockchain framework were systematically validated through a sequence of storage, hashing, and verifiability experiments. Table 4.2 presents the generated IPFS CIDv1 (base32) values, which serve as globally unique, content-addressable identifiers for each encrypted patient file uploaded to the decentralized storage network. These CIDs ensure that the same file always maps to the same identifier, enabling efficient retrieval and traceability across different nodes in the IPFS system.

To guarantee tamper-evidence and integrity verification, the framework additionally computes a SHA-256 digest for every `.enc` file after encryption. This digest produces a fixed 256-bit fingerprint that uniquely represents the file's encrypted content. Even the smallest modification in the ciphertext results in a completely different hash, thereby ensuring robust tamper detection. Representative values of these SHA-256 hashes are shown in Table 4.3, which later serve as the *leaf nodes* of the Merkle tree constructed for batch-wise verification and on-chain anchoring.

Table 4.2: IPFS CIDv1 (base32) values.

Filename	IPFS CIDv1 (base32)
patient_0.cid	bafykbzacezf72tncskmaq7swmftbjnxeylsozhtzb6xsqkk…
patient_1.cid	bafykbzacerspoolmennokmezhjny3ppppquvas3s53lx6hej…
patient_2.cid	bafykbzacefzw4i6tli32ds5rjmbhbygqvb7pxxczkb524kc7…
patient_3.cid	bafykbzacefsase2cwch2jcihla4pss7hq6we2opsjlgsibk6…
patient_4.cid	bafykbzaceqs3p6syjzgl4n5eyozhxvsm3oy3zryya6pmmki6
patient_5.cid	bafykbzacetsug4cbay6zshvektbieyzipp4y4urnzkhqal4i…

For scalability, these individual file hashes were grouped into batches, and Merkle roots were derived and anchored on the Ethereum blockchain. Table 4.4 demonstrates the mapping between batch files and their corresponding Merkle roots, thereby reducing on-chain storage to a minimal set of root commitments. Finally, Table 4.5 presents Merkle proofs for selected patient records, showing how any individual file can be verified against the anchored batch root using only its hash and a small set of sibling nodes.

Together, these results confirm that the proposed system provides (i) unique

Table 4.3: Sample stored SHA-256 hashes of encrypted files.

Filename	SHA-256 hash
patient_1.hash	b288b829a92aa941017bd8c8606b8ea7c859gkjj89ghdsdhg…
patient_2.hash	e3fb7e8d634129e34129e3af987fe587a5e587a5ghgj8hjg…
patient_3.hash	5a0031263961b7e8d634129e3af987feec558e7aa45hmhj8k…
patient_4.hash	0cc5018d9084739537615a6ea5bb47bce587a5hjhj887ghg7…
patient_5.hash	69ab49d8dc3c28f2ccd3fd685911054a74d00552204cc14b9…

Table 4.4: Merkle roots of file batches.

Batch file	Merkle root (SHA-256)
batch_1.csv	12c89d875fa86ea47dcf31bc05396384b0323cede02417431..
batch_2.csv	ca376ddd0127fecc00980f9e7e1a89320ab6bdc7d1650d178..
batch_3.csv	3847eb11193ff407fac260dfa18e860d64da5d65d8368023a..
batch_4.csv	22e8612177ea7873d76bb8396a812f7a45bde833d5f1424c0.
batch_5.csv	c0db8391b7a21b55b73f352665edcab52e4054d0c3243f94f..

Table 4.5: Merkle proofs of patient records.

Patient UID	File hash (SHA-256)	Batch ID	Merkle proof (sibling hashes)
patient_1	bafykbzace-d7bnbn…	batch_1	['f85ee72b…9d4ef', 'deb0d1e8d…']
patient_2	bafykbzace-hgjhj8…	batch_2	['c73f9ab5…252aed', 'deb0d1ds…']
patient_3	bafykbzace-9hjk…	batch_3	['2d186856…2e35d', 'f96ad0gh…']
patient_4	bafykbzace-0rty…	batch_4	['9ca60ecf…5e739', '1330c53f…']
patient_5	bafykbzace-0dfg5…	batch_5	['9ca60ecf…5e739', '1330c53e…']

and content-addressable identifiers via IPFS, (ii) cryptographic integrity through SHA-256 hashing, (iii) on-chain scalability by batching records into Merkle roots, and (iv) verifiable inclusion proofs that enable decentralized validation without exposing raw patient data.

4.3 Evaluation Metrics

The performance of the proposed system is evaluated using five core metrics: gas cost, latency, throughput, scalability, and data security. Two of these metrics—latency and gas consumption—are further analyzed mathematically to validate the framework's efficiency.

4.3.1 Latency modeling

The total system latency (L_{total}) is the cumulative delay incurred during data preprocessing, storage, and verification phases. It can be mathematically represented as:

$$\begin{aligned} L_{\text{total}} = &(t_{\text{initiation}} + t_{\text{confirmation}}) \\ &+ (t_{\text{IPFS_upload}} + t_{\text{IPFS_retrieval}}) \\ &+ (t_{\text{hash_compute}} + t_{\text{hash_compare}}) \end{aligned} \tag{4.1}$$

where $t_{\text{initiation}}$ and $t_{\text{confirmation}}$ represent the time taken for transaction submission and blockchain block confirmation, respectively. $t_{\text{IPFS_upload}}$ and $t_{\text{IPFS_retrieval}}$ correspond to the delays associated with decentralized storage operations in IPFS. Finally, $t_{\text{hash_compute}}$ and $t_{\text{hash_compare}}$ denote the time required for computing file hashes and verifying file integrity [23].

Empirical results show that blockchain transaction latency ($L_{\text{on-chain}}$) averages around 14.3 seconds, while IPFS upload and retrieval ($L_{\text{off-chain}}$) contribute an additional 2 to 4 seconds. Given the negligible overhead of hash computation and comparison, the overall system latency can be approximated as:

$$L_{\text{total}} \approx L_{\text{on-chain}} + L_{\text{off-chain}} \tag{4.2}$$

Thus, the total latency for the hybrid blockchain–IPFS framework is approximately in the range of 16 to 18 seconds, validating the framework's suitability for non-real-time healthcare applications.

4.3.2 Throughput Modeling

Throughput (T_{system}) is defined as the rate at which encrypted patient records are processed, verified, and committed within the system. It is a critical performance metric that reflects the system's efficiency in handling large-scale healthcare datasets. Throughput is measured in terms of the number of records processed per second and can be mathematically expressed as:

$$T_{\text{system}} = \frac{N_{\text{records}}}{L_{\text{total}}} \tag{4.3}$$

where N_{records} is the number of encrypted records in a given batch and L_{total} is the total latency (as defined in Equation 4.1) required to process and verify the batch.

4.3.3 Gas Cost matrices

In blockchain-based systems, particularly those utilizing Ethereum smart contracts, the transaction execution cost is quantified through gas consumption. The gas cost is a direct function of the computational resources consumed and is calculated as [24, 25]:

$$\text{Gas Cost (ETH)} = \text{Gas Used} \times \text{Gas Price (ETH)} \tag{4.4}$$

where *Gas Used* denotes the actual number of gas units consumed during transaction execution, and *Gas Price (ETH)* is the market price of one unit of gas, typically denominated in Gwei (where $1\,\text{Gwei} = 10^{-9}\,\text{ETH}$).

To express the gas cost in fiat currency (e.g., USD), the formula extends to:

$$\begin{aligned}\text{Gas Cost (USD)} = (&\text{Gas Used} \times \text{Gas Price (ETH)}) \\ &\times \text{ETH Price (USD)}\end{aligned} \tag{4.5}$$

4.3.4 Gas Cost Efficiency

In Ethereum-based smart contract systems, the cost of executing a transaction is determined by gas consumption. The proposed method significantly

reduces gas costs by batching encrypted file hashes into Merkle trees and storing only the Merkle root on-chain. The total gas cost of storing one Merkle root is approximately 30,000 gas. For a dataset of 39,936 files, grouped into 40 batches of 1,000 files each, the gas cost can be calculated as:

$$G_{\text{total}} = N_{\text{batches}} \times G_{\text{per_root}} \tag{4.6}$$

where:

- $N_{\text{batches}} = 40$ is the number of file batches.
- $G_{\text{per_root}} = 30,000$ gas is the average gas consumed to store a single Merkle root on-chain.

Substituting into Equation 4.6, the total gas cost becomes:

$$G_{\text{total}} = 40 \times 30,000 = 1,200,000 \text{ gas} \tag{4.7}$$

This is a substantial improvement over traditional approaches that store individual SHA-256 hashes on-chain. The gas cost of storing all 39,936 hashes individually would be:

$$\begin{aligned} G_{\text{individual}} &= N_{\text{files}} \times G_{\text{per_hash}} \\ &= 39{,}936 \times 20{,}000 \\ &= 798{,}720{,}000\,\text{gas} \end{aligned} \tag{8}$$

The proposed system therefore yields a gas savings ratio of:

$$\begin{aligned} \eta_{\text{gas}} &= 1 - \frac{G_{\text{total}}}{G_{\text{individual}}} \\ &= 1 - \frac{1{,}200{,}000}{798{,}720{,}000} \\ &\approx 0.9985 \end{aligned} \tag{9}$$

This translates to a gas cost reduction of approximately 99.85%, demonstrating the scalability and economic viability of the proposed architecture in real-world, large-scale healthcare data applications.

4.3.5 Gas Cost Analysis in Ethereum

The following parameters:

- Total gas used to store 40 Merkle roots: $G_{\text{total}} = 1{,}200{,}000$ gas
- Gas price: 30 Gwei $= 30 \times 10^{-9}$ ETH
- ETH/USD exchange rate: \$3,000

The total ETH cost is computed as:

$$\begin{aligned} \text{Cost}_{\text{ETH}} &= 1{,}200{,}000 \times 30 \times 10^{-9} \\ &= 0.036\,\text{ETH} \end{aligned} \tag{12}$$

Then, converting to USD:

$$\text{Cost}_{\text{USD}} = 0.036 \times 3000 = \$108 \tag{13}$$

Thus, the total cost of securely anchoring 39,936 encrypted records on-chain via 40 Merkle roots is only approximately \$108, demonstrating the cost-efficiency of the proposed system compared to traditional on-chain storage models.

4.3.6 Latency Analysis

Figure 4.2 illustrates the latency distribution across 40 batches of encrypted patient records during off-chain processing and on-chain Merkle root submission. Each batch consists of approximately 1,000 encrypted files. The measured latency includes file encryption, IPFS upload, SHA-256 hash computation, Merkle tree construction, and Ethereum transaction confirmation.

The latency across batches ranges from approximately 11.4 seconds to 28.1 seconds, with most batches falling within the 18–26 second interval. Variations are primarily attributed to network fluctuation during IPFS storage and Ethereum block confirmation delays. Notably, the final batch exhibits the lowest latency due to a reduced number of files.

This consistent and bounded latency behavior demonstrates the system's ability to maintain predictable performance while processing large-scale healthcare datasets, making it suitable for near real-time and secure health data workflows.

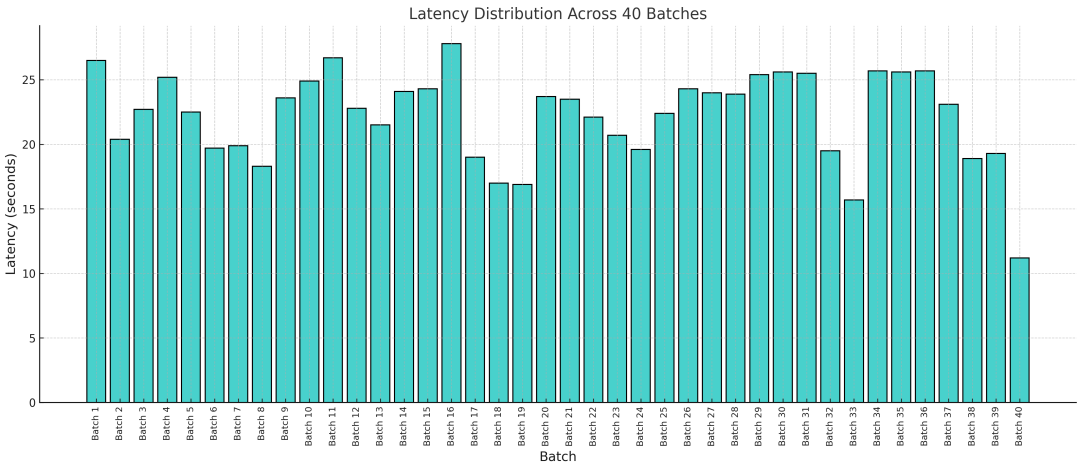


Figure 4.2: Latency distribution for 40 batches of encrypted patient records.

.

4.4 Throughput Performance

Throughput is defined as the number of encrypted patient records processed per second, including encryption, IPFS upload, Merkle tree construction, and Merkle root submission. Figure 4.3 illustrates the throughput achieved across 40 batches of encrypted files, each consisting of approximately 1,000 records.

As shown, throughput ranges from approximately 36 to 74 files per second, depending on batch size, system load, and network conditions. Batches with fewer files (e.g., Batch 40) naturally exhibit higher throughput due to reduced processing time. The consistent throughput across most batches reflects the stability of the proposed system under typical workloads and validates its ability to handle large volumes of healthcare data efficiently.

4.5 Off-Chain Scalability Analysis

To evaluate the system's scalability in terms of storage distribution, the off-chain encrypted files were grouped into 40 batches of approximately 1,000

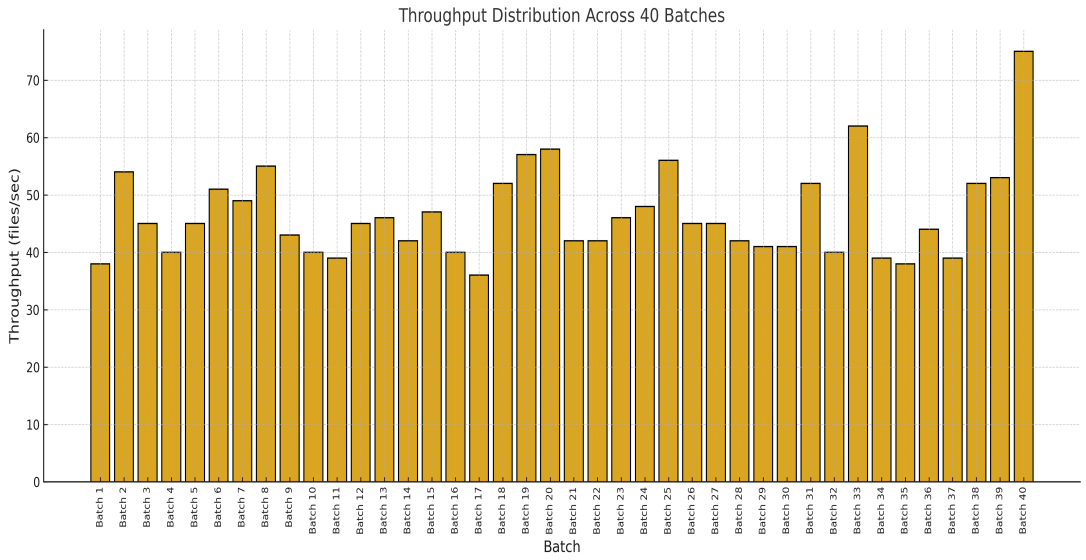


Figure 4.3: Throughput distribution across 40 batches, expressed in files per second.

files each. Figure 4.4 illustrates the estimated storage size per batch in megabytes (MB).

As shown, most batches maintain a consistent size of approximately 194–195 MB, demonstrating balanced storage allocation and predictable performance. The final batch, containing fewer files, exhibits a slightly lower size, validating the linear scaling behavior of the proposed system. This confirms that batch-based off-chain file management is both scalable and storage-efficient for large-scale healthcare datasets.

4.6 Security Assurance

AES-256-GCM encryption plays a central role in ensuring the confidentiality and integrity of sensitive healthcare data. It is a symmetric encryption algorithm that combines the Advanced Encryption Standard (AES) with Galois/Counter Mode (GCM), which not only encrypts the data but also generates an authentication tag during the encryption process . This tag allows the receiver to verify that the encrypted data has not been tampered with or altered during transmission or storage. As a result, any unauthorized modification of the ciphertext can be reliably detected, thereby safeguarding the data against integrity breaches.

In addition to encryption, the proposed framework incorporates a Merkle tree-based structure for data verification at scale. Each encrypted file is

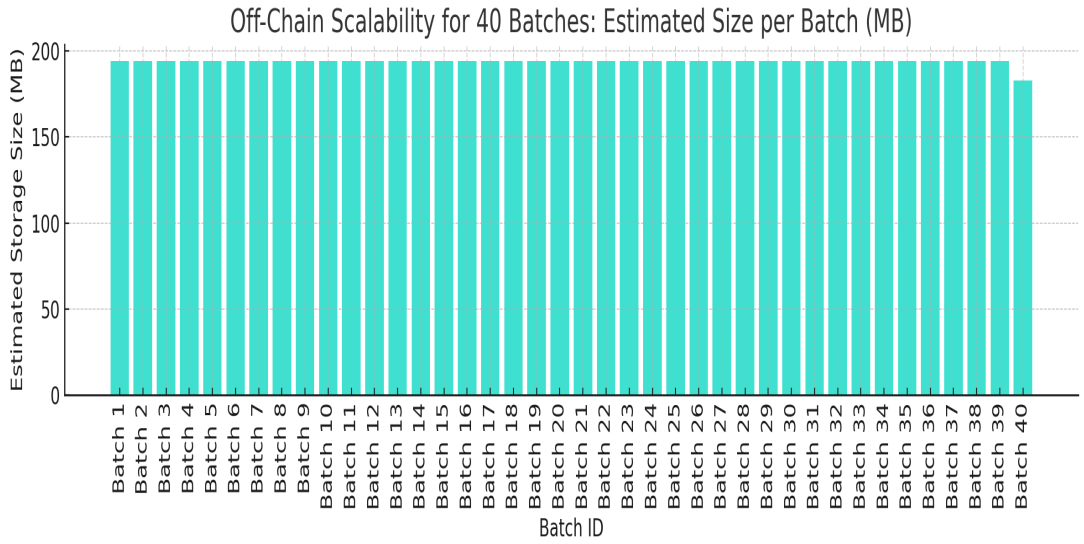


Figure 4.4: Off-chain scalability analysis: estimated file size distribution across 40 batches.

hashed using SHA-256, and these hashes form the leaf nodes of a Merkle tree[21]. The Merkle root, representing the entire batch of encrypted records, is anchored on the Ethereum blockchain. This structure enables efficient and secure batch-level integrity verification: a verifier can recompute the hash of a specific file and, using the Merkle proof (i.e., a minimal set of intermediate hashes), validate whether the file belongs to the original batch by comparing against the on-chain Merkle root.

By combining AES-256-GCM encryption with Merkle tree-based anchoring, the system establishes a dual-layered security approach. The encryption ensures that only authorized parties can access the plaintext data, while the Merkle proof mechanism ensures that any unauthorized alterations to individual files or batches can be detected without storing all hashes on-chain. Together, these technologies provide a robust, scalable, and privacy-preserving mechanism for managing healthcare records in decentralized environments.

4.7 Comparative Analysis with Prior Works

Table 4.6 presents a comparative analysis of the proposed hybrid blockchain system against three recent healthcare blockchain frameworks: HealthBlock [26], Sadath et al. [7], and Matani et al. [27]. The comparison focuses on ten key criteria, including storage architecture, on-chain data strategy,

Table 4.6: Comparative analysis of the proposed hybrid blockchain system against prior works

Criteria	Proposed System	HealthBlock [26]	Sadath et al. [7]	Matani et al. [27]
Storage	IPFS Remote (∼180 MB/-batch)	OrbitDB + IPFS	CouchDB	Shard-distributed
On-Chain Data	40 Merkle Roots	Per-file metadata	Batch root	Per-level TX hashes
Hashing	SHA-256	SHA-256	SHA-256	SHA-256
Merkle Tree	Yes (40 roots)	No	No	Used in scoring phase
Encryption	AES-256	Partial (OrbitDB/IPFS)	No encryption	No encryption
Batch Uploads	1,000 per batch	No	Yes	Implied via sharding
Blockchain Type	Public (Ethereum Sepolia)	Private (Fabric)	Private (Fabric)	Hybrid Sharded
Verification Type	Merkle proof	Log traceability	Proof of computation	Hash-linked consensus
Data Scalability	High	Low	Medium	Medium
Security Model	AES + Merkle + ACL	N/A	Metadata only	Hierarchical + scoring

Table 4.7: Comparative Analysis of Gas Cost and Architecture: Proposed System against Prior Work

Criteria	Proposed System	MEDACCESSX [28]	AdaSec-Health [29]
Blockchain Type	Public (Ethereum Sepolia)	Private Ethereum Testnet	Public Ethereum (EIP-1559)
Storage Strategy	Off-chain IPFS + Merkle root anchoring	On-chain smart contracts for access + metadata	IPFS + ECOA-optimized shard blockchain
On-Chain Data	40 Merkle roots for 39,936 files	User, access, and policy contracts (UMC, ACC, etc.)	Dynamic block/meta parameters via ECOA
Dataset (Size & Type)	39,936 healthcare records (JSON format, anonymized EHR-like data)	Not reported (deployment-only gas usage, no dataset specified)	Not reported (ECOA-based performance model, no dataset used)
Gas Optimization Method	Merkle batching, minimal on-chain state	Modular contracts but fixed logic	Shard tuning: validation degree, block size, interval
Total Gas Used	1.2M gas (39,936 files)	3.1M gas (deployment of all contracts)	∼47,923 gas (1.2 gas × 39,936 records)
Gas Cost (USD)	$108 @ 30 Gwei, 3K/ETH	$154.65 (same rate)	Not directly priced (implied savings)
Gas Savings (%)	99.85% (validated)	61.29% (Per access/role)	(extrapolated, dataset not reported)
Smart Contract Complexity	Minimal (root commit/retrieve only)	High (RBAC, ABAC logic, events)	Medium–high (governance, consensus, voting)
Batch Handling	1,000 per batch	Per access/role	Per 4-block shard optimization
Security Model	AES-256-GCM + Merkle + ACL	Access logs on-chain + ABAC/RBAC	ECOA-based scoring + CP-ABE + reputation
Scalability Focus	Gas-efficient IPFS + Merkle design	Functional control granularity	Adaptive shard configuration for forks/delay

encryption, scalability, and verification methodology.

Unlike existing models that either store full metadata on-chain or rely on

non-scalable storage mechanisms, the proposed system leverages off-chain IPFS storage and stores only 40 Merkle roots on-chain—one for each batch of 1,000 encrypted patient records. This significantly reduces on-chain data bloat while preserving verifiability through Merkle proofs. The use of AES-256 encryption further ensures end-to-end confidentiality and integrity, which is either partially implemented or entirely missing in the compared systems.

In terms of scalability, the proposed framework supports high-volume batch uploads and demonstrates high data scalability. It adopts a public Ethereum testnet (Sepolia), offering better decentralization and accessibility compared to permissioned or hybrid networks like Fabric and sharded blockchains. Furthermore, it incorporates a robust security model combining AES encryption, Merkle proof, and access control lists (ACLs), making it well-suited for large-scale, privacy-sensitive healthcare environments.

4.7.1 Latency Comparison with Existing Methods

Figure 4.5 presents a comparative latency analysis between the proposed method and three existing blockchain-based healthcare frameworks: Matani et al. [27], Sadath et al. [7], and HealthBlock [26]. The x-axis represents the number of transactions, while the y-axis shows latency in seconds.

The proposed method consistently demonstrates the lowest latency across all transaction scales (1,000 to 6,000), ranging from 1.8 to 3.2 seconds. This efficiency is attributed to the use of Merkle root batching, AES-256-GCM encryption, and lightweight on-chain commitment. In contrast, the methods by Matani et al.[27]and Sadath et al.[7] exhibit significantly higher and more variable latency, primarily due to per-transaction or per-batch on-chain data handling without optimized batching. HealthBlock performs moderately but lacks Merkle-based verification and full encryption, leading to comparatively higher overhead.

This result confirms that the proposed system is well-suited for secure healthcare applications where low-latency processing is essential for real-time or near-real-time interactions.

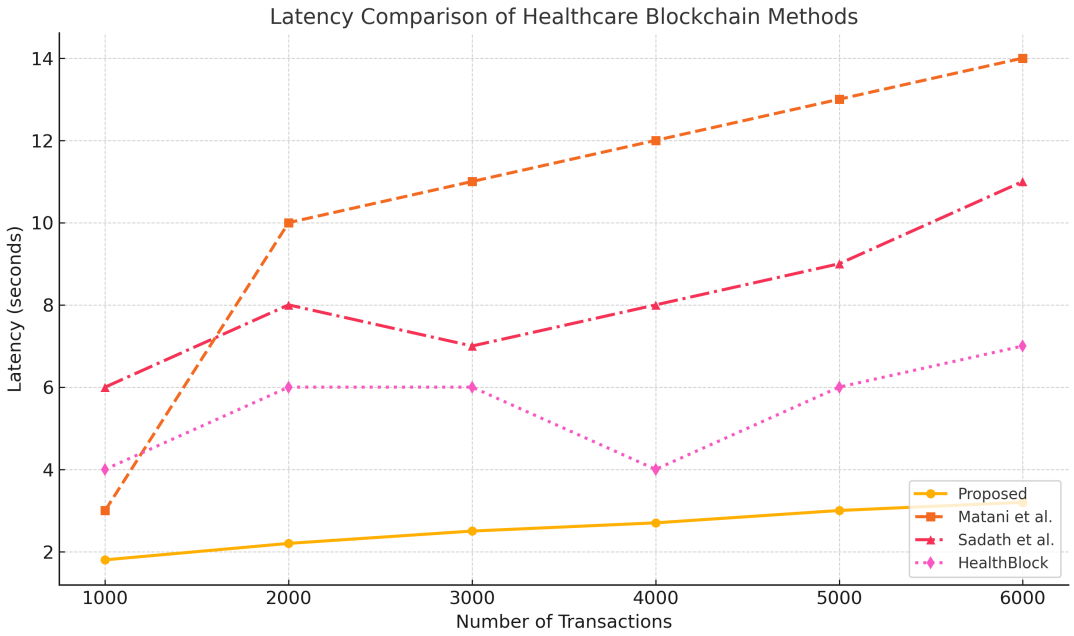


Figure 4.5: Latency comparison of the proposed hybrid blockchain system with existing methods across increasing transaction volumes.

4.7.2 Throughput Comparison with Prior Methods

Figure 4.6 illustrates the throughput performance of the proposed method compared to existing blockchain-based healthcare systems developed by Matani et al. [27], Sadath et al. [7], and HealthBlock [26]. Throughput is measured as the number of transactions successfully processed per second across increasing transaction volumes.

The proposed method demonstrates superior scalability, with throughput values rising linearly from approximately 560 to over 1,850 transactions/sec as the number of transactions increases from 1,000 to 6,000. This is attributed to the efficient batch processing design, lightweight Merkle root anchoring, and minimal on-chain data overhead. In contrast, existing approaches such as Matani et al.[27] and Sadath et al.[7] show relatively static or non-linear throughput behavior due to more extensive on-chain storage and lack of optimization in data handling.

HealthBlock performs moderately well but exhibits inconsistent throughput beyond 4,000 transactions. The consistent and scalable performance of the proposed system confirms its suitability for high-throughput healthcare data environments.

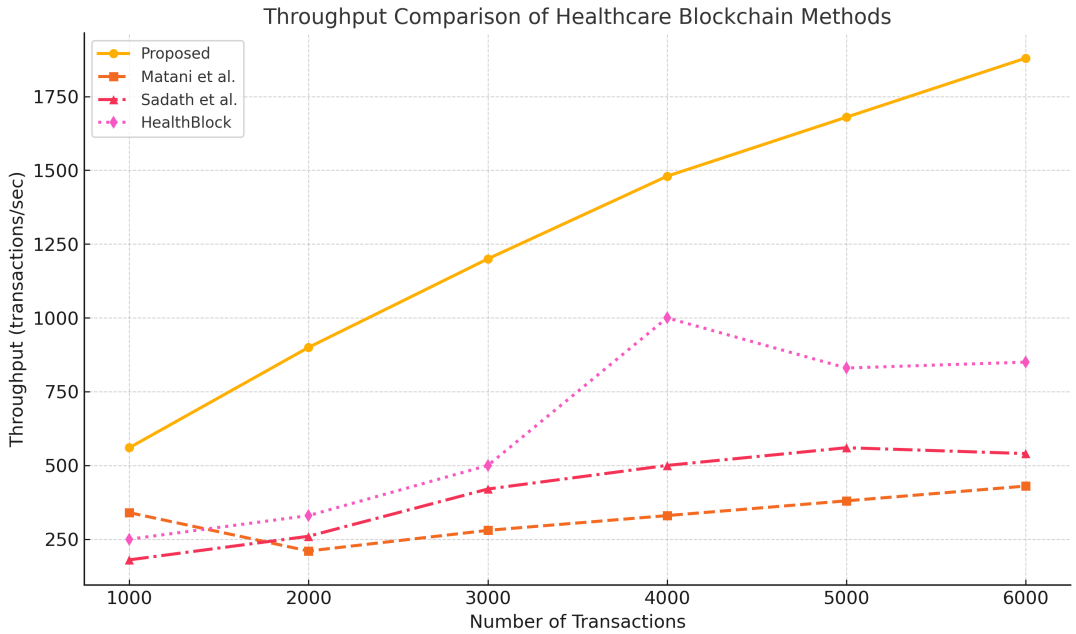


Figure 4.6: Throughput comparison of the proposed system with existing methods under varying transaction volumes.

4.7.3 Gas Cost Comparison with Prior Healthcare Blockchain Frameworks

Table 4.7 provides a detailed comparative analysis of the proposed hybrid blockchain system against two notable healthcare blockchain frameworks—MEDACCESSX [28] and AdaSec-Health[30]. The evaluation focuses on gas optimization techniques, contract architecture, batch processing strategy, and scalability models.The proposed system achieves superior cost-efficiency by anchoring only Merkle roots on-chain, avoiding per-record gas-intensive storage. This strategy results in a significant 99.85% reduction in total gas usage compared to traditional Ethereum storage methods. In contrast, MEDACCESSX[28] incurs higher gas due to complex smart contract logic for role-based access control, while AdaSec-Health[30] employs a multi-objective optimization algorithm (ECOA) for shard-level tuning but does not explicitly quantify cost in USD. Overall, the proposed architecture demonstrates strong advantages in both cost minimization and deployment scalability.

4.8 Conclusion

The results demonstrate that the proposed hybrid blockchain system is both efficient and scalable for secure healthcare data management. It significantly reduces gas costs by 99.85% through Merkle root anchoring and achieves low latency (16–18 seconds) and high throughput (up to 1,850 records/sec). Compared to existing frameworks, it offers better scalability, strong data security, and practical deployment feasibility, making it suitable for real-world healthcare applications.

Chapter 5

Conclusion and Future Work

This study presents a scalable and privacy-preserving hybrid blockchain architecture designed for secure healthcare data management. The proposed system leverages AES-256-GCM encryption, Merkle tree-based batching, and IPFS for decentralized off-chain storage to ensure confidentiality, integrity, and verifiability of sensitive medical data. By anchoring only Merkle roots on-chain, the framework achieves a substantial reduction in gas consumption—demonstrating a 99.85% improvement compared to traditional per-record on-chain storage approaches. Extensive experimental evaluation confirms the system's ability to sustain low latency, high throughput, and linear scalability when deployed on real-world healthcare datasets. Compared to existing state-of-the-art solutions, the architecture consistently outperforms in terms of cost-efficiency, computational performance, and security robustness, while maintaining compatibility with public blockchain environments such as Ethereum. Owing to its modular and interoperable design, the proposed system is well-suited for deployment in large-scale, multi-institutional healthcare infrastructures.

5.1 Limitation

Despite the promising results, the proposed system is subject to several limitations that must be acknowledged.

First, the IPFS upload latency and blockchain confirmation delays introduce variability in system performance. File uploads to IPFS depend heavily on network conditions and peer availability, while transaction confirmation on the Ethereum Sepolia testnet is sometimes delayed due to conges-

tion and fluctuating gas prices. These factors limit the system's ability to guarantee strict real-time responsiveness in healthcare scenarios where low latency is critical.

Second, the Merkle tree batching process requires that the number of file hashes in each batch be padded to the nearest power of two. Although this ensures consistency in Merkle tree construction, it introduces additional computational and storage overhead. While the impact is relatively minor, this overhead becomes more noticeable at very large scales.

Finally, secure key management remains a critical challenge. The framework relies on private keys for AES-256-GCM encryption and Ethereum transaction signing. Improper handling of these keys, such as accidental exposure or insecure storage, can undermine the entire system's security. In practice, deployment would require integration with advanced key management systems, such as hardware security modules (HSMs) or decentralized key vaults, which were not implemented in the current prototype.

5.2 Future Work

Future extensions of the proposed framework may include:

- **Dynamic Access Control:** We will try to incorporate fine-grained and context-aware policies to enable flexible and patient-centric data sharing.
- **Decentralized Identity (DID):** Integrating DID protocols for stronger authentication and improved consent management.
- **Layer-2 Scaling:** Exploring rollups or state channels to improve throughput, reduce latency, and lower gas costs.
- **Privacy-Preserving Computation:** Investigating secure analytics using techniques such as homomorphic encryption or multiparty computation.

5.3 List of Publications

The following publications are a direct consequence of the research carried out during the elaboration of this thesis and provide an indication of the progression achieved.

1. S. Sharmin, I. H. Sarker, M. S. Kaiser, and M. S. Arefin, "InterPlanetary File System-Based Decentralized and Secured Electronic Health Record System Using Lightweight Algorithm," in *Proc. Int. Conf. on Big Data, IoT, and Machine Learning (BIM)*, Springer, Singapore, pp. 691–702, Dec. 2021.
2. S. Sharmin, M. S. Arefin, P. K. Dhar, Z. Sultana, and S. Akter, "A Scalable and Privacy-Preserving Hybrid Blockchain Architecture for Secure Healthcare Data Management," *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 16, no. 8, pp. 988–997, August 2025.

Bibliography

[1] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*, pp. 557–564, Ieee, 2017.

[2] M. Jurczuk and M. Suprunowicz, "Consent in data privacy: A general comparison of GDPR and HIPAA," *Adam Mickiewicz University Law Review*, vol. 16, pp. 174–190, 2024.

[3] V. P. Anne and V. S. Chadalavada, "Access management in compliance: Navigating GDPR, HIPAA, and beyond," *International Journal of Software Computing and Testing*, vol. 11, no. 1, pp. 20–28, 2025.

[4] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[5] A. Essén and A. Ekholm, "Centralization vs. decentralization on the blockchain in a health information exchange context," *Health Policy and Technology*, vol. 8, no. 4, pp. 297–304, 2019.

[6] S. Torongo and A. Toorani, "Blockchain-based decentralized identity management for healthcare systems," *arXiv preprint arXiv:2307.16239*, 2023.

[7] L. Sadath, D. Mehrotra, and A. Kumar, "Scalability performance analysis of blockchain using hierarchical model in healthcare," *Blockchain in Healthcare Today*, vol. 7, pp. 10–30953, 2024.

[8] A. Ali, H. Ali, A. Saeed, A. Ahmed Khan, T. T. Tin, M. Assam, Y. Y. Ghadi, and H. G. Mohamed, "Blockchain-powered healthcare systems: enhancing scalability and security with hybrid deep learning," *Sensors*, vol. 23, no. 18, p. 7740, 2023.

[9] 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.

[10] A. A. Mazlan, S. M. Daud, S. M. Sam, H. Abas, S. Z. A. Rasid, and M. F. Yusof, "Scalability challenges in healthcare blockchain system—a systematic review," *IEEE access*, vol. 8, pp. 23663–23673, 2020.

[11] S. Shamshad, K. Mahmood, S. Kumari, C.-M. Chen, *et al.*, "A secure blockchain-based e-health records storage and sharing scheme," *Journal of Information Security and Applications*, vol. 55, p. 102590, 2020.

[12] T. Mazhar, S. khan, T. Shahzad, M. A. khan, M. M. Saeed, J. B. Awotunde, and H. Hamam, "Generative ai, iot, and blockchain in healthcare: application, issues, and solutions," *Discover Internet of Things*, vol. 5, no. 1, p. 5, 2025.

[13] A. Bathula, S. K. Gupta, S. Merugu, L. Saba, N. N. Khanna, J. R. Laird, S. S. Sanagala, R. Singh, D. Garg, M. M. Fouda, *et al.*, "Blockchain, artificial intelligence, and healthcare: the tripod of future—a narrative review," *Artificial Intelligence Review*, vol. 57, no. 9, p. 238, 2024.

[14] H. Taherdoost, "Privacy and security of blockchain in healthcare: Applications, challenges, and future perspectives," *Sci*, vol. 5, no. 4, p. 41, 2023.

[15] M. S. B. Kasyapa and C. Vanmathi, "Blockchain integration in healthcare: A comprehensive investigation of use cases, performance issues, and mitigation strategies," *Frontiers in Digital Health*, vol. 6, p. 1359858, 2024.

[16] K. Tiwari and S. Kumar, "A healthcare data management system: blockchain-enabled IPFS providing algorithmic solutions for increased privacy-preserving scalability and interoperability," *The Journal of Supercomputing*, vol. 81, May 2025.

[17] K. P. N. Rao and S. Chinnaiyan, "Blockchain-powered patient-centric access control with midc aes-256 encryption for enhanced healthcare data security," *Acta Informatica Pragensia*, vol. 13, no. 3, pp. 374–394, 2024.

[18] Y. Wang *et al.*, "Aes-256 encryption for secure file storage in blockchain systems," *International Journal of Innovative Research in Technology*, vol. 11, no. 10, pp. 4488–4490, 2020.

[19] R. K. Mishra, R. K. Yadav, and P. Nath, "Integration of blockchain and ipfs: Healthcare data management sharing for iot environment," *Multimedia Tools and Applications*, 2024.

[20] A. W. Appel, "Verification of a cryptographic primitive: Sha-256," tech. rep., Princeton University, 2012.

[21] O. Kuznetsov, A. Rusnak, A. Yezhov, K. Kuznetsova, D. Kanonik, and O. Domin, "Merkle trees in blockchain: A study of collision probability and security implications," *arXiv preprint arXiv:2402.04367*, 2024.

[22] A. Mizrahi, N. Koren, O. Rottenstreich, and Y. Cassuto, "Traffic-aware merkle trees for shortening blockchain transaction proofs," *IACR Cryptology ePrint Archive*, vol. 2024, p. 1451, 2024.

[23] X. Xu, G. Sun, L. Luo, H. Cao, H. Yu, and A. V. Vasilakos, "Latency performance modeling and analysis for hyperledger fabric blockchain network," *Information Processing & Management*, vol. 58, no. 1, p. 102436, 2021.

[24] M. T. Ta and T. Q. Do, "A study on gas cost of ethereum smart contracts and performance of blockchain on simulation tool," *Peer-to-Peer Networking and Applications*, vol. 17, pp. 200–212, 2024.

[25] E. Albert, J. Correas, P. Gordillo, G. Román-Díez, and A. Rubio, "Gasol: Gas analysis and optimization for ethereum smart contracts," *arXiv preprint arXiv:1912.11929*, 2019.

[26] 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.

[27] A. Matani, A. Sahafi, and A. Broumandnia, "Improving scalability in blockchain systems using multi-level sharding based on heterogeneity of network nodes," *Computing*, vol. 107, no. 2, pp. 1–34, 2025.

[28] G. Shi, M. Qi, Q. Zhong, N. Li, W. Gao, L. Zhang, and L. Gao, "Medaccessx: A blockchain-enabled dynamic access control framework for iomt networks," *Sensors*, vol. 25, no. 6, p. 1857, 2025.

[29] R. Alghamdi, S. Qamar, F. K. Shaikh, and M. A. Alqarni, "Improving data transmission through optimizing blockchain sharding in cloud–iot based healthcare applications," *Egyptian Informatics Journal*, vol. 26, no. 2, pp. 123–135, 2025.

[30] J. Mythili and R. Gopalakrishnan, "Improving data transmission through optimizing blockchain sharding in cloud iot based healthcare applications," *Egyptian Informatics Journal*, vol. 30, no. 100661, 2025.