Nexa: A Blockchain Architecture for Secure and Scalable EHR Solutions—Balancing Blockchain Trilemma, with a Scalable Public Blockchain and Threshold Cryptography

Ahmed Abbasi * and Mohamed Nour Humeidi *

Posted Date: 5 February 2026

doi: 10.20944/preprints202602.0280.v1

Keywords: blockchain; EHRs (electronic health records); Avalanche; distributed threshold cryptography; smart contracts; cryptography; scalability; security

 doi:10.20944/preprints202602.0280.v1

Article

Nexa: A Blockchain Architecture for Secure and Scalable EHR Solutions—Balancing Blockchain Trilemma, with a Scalable Public Blockchain and Threshold Cryptography

Ahmed Abbasi * and Mohamed Nour Humeidi *

Kristianstad University, SE-291 88 Kristianstad

* Correspondence: ahmed.abbasi0015@stud.hkr.se (A.A.); mohamed_nour.humeidi0008@stud.hkr.se (M.N.H.)

Abstract

This thesis introduces Nexa, a blockchain architecture designed to balance the critical trade-offs between security, scalability, and decentralization in Electronic Health Record (EHR) systems. To address the blockchain trilemma (RQ1), Nexa strategically combines Avalanche's high-performance public blockchain with decentralized IPFS storage, optimizing for the high throughput and low finality required in healthcare. To mitigate risks from centralized key management (RQ2), Nexa implements a distributed threshold cryptography scheme using Elliptic Curve Diffie-Hellman (ECDH), ChaCha20-Poly1305, and Shamir's Secret Sharing (SSS). This design leverages smart contract-based access control and oracle-assisted decryption to enhance security without a central point of failure. Performance benchmarks on the Avalanche testnet, using synthetic datasets, validate this balanced approach, demonstrating efficient operation latencies and minimal costs suitable for clinical workflows while ensuring patient control.

Keywords: blockchain; EHRs (electronic health records); Avalanche; distributed threshold cryptography; smart contracts; cryptography; scalability; security

---

Terminology

- **Blockchain:** Distributed digital ledger that records transactions across multiple computers securely.
- **Transaction:** Record of value transfer or state change on a blockchain network.
- **Blockchain Trilemma:** The challenge of balancing security, scalability, and decentralization in blockchain systems.
- **Public Blockchain:** A permissionless network where anyone can participate and validate transactions.
- **Smart Contract:** Self-executing code that enforces agreement terms on a blockchain.
- **EVM (Ethereum Virtual Machine):** Runtime environment for executing smart contracts on Ethereum and compatible networks.
- **Gas:** Computational fee for executing operations on blockchain networks.
- **Latency:** Time delay between initiating a blockchain transaction and its finality.
- **TPS (Transactions Per Second):** A measure of blockchain processing capacity.
- **Avalanche:** A blockchain platform for sub-second finality and high throughput.
- **Oracle:** Service connecting smart contracts to external live data sources.
- **Threshold Cryptography:** Technique requiring multiple parties to collaborate for cryptographic operations.
- **Distributed Threshold Cryptography:** Distributed implementation of threshold cryptography without central authority.

 doi:10.20944/preprints202602.0280.v1

- **SSS (Shamir's Secret Sharing):** Algorithm dividing secrets into parts requiring a threshold number to reconstruct.
- **ECDH (Elliptic Curve Diffie-Hellman):** Cryptographic key exchange method using Elliptic Curve mathematics for secure communication.
- **ChaCha20-Poly1305:** Symmetric encryption algorithm combining stream cipher with message authentication code.
- **Decentralized Storage:** Data storage across multiple nodes without central control.
- **IPFS (InterPlanetary File System):** Distributed P2P protocol creating content-addressable file system.
- **RBAC (Role-based access control):** Access control method based on users' organizational roles.
- **EHR (Electronic Health Record):** Digital version of patient medical records containing health information.
- **On-chain / Off-chain:** A state that is intended to be maintained within (or outside of) a blockchain network.

1. Introduction

1.1. Context and Motivation

The healthcare industry is facing a growing crisis of data security, with cyberattacks on hospitals and clinics becoming increasingly frequent and sophisticated. A single breach can compromise thousands of patient records, disrupt critical services, and cost healthcare organizations millions of dollars in fines and remediation efforts [1]. Traditional Electronic Health Record (EHR) systems are often vulnerable to these attacks due to centralized architecture, inadequate access controls, and a lack of strong security protocols.

Blockchain technology offers a potential solution to these challenges by providing a decentralized, tamper-proof, and auditable environment for managing sensitive medical data. Its inherent security features, such as data integrity (ensured through cryptographic hashing) and decentralized consensus mechanisms, can help prevent unauthorized access, ensure data integrity, and streamline regulatory compliance. However, the widespread adoption of blockchain in healthcare depends on overcoming its scalability limitations. Traditional blockchain systems often prioritize security and decentralization at the expense of transaction speed and efficiency, which introduces a significant barrier to EHR systems that require low-latency (typically defined as response times within milliseconds to a few seconds) access to medical data across hospitals and clinics.

This thesis addresses this challenge by introducing Nexa, a distinct blockchain-based EHR architecture that attempts to find a balance between scalability and security without compromising the decentralization aspect. By utilizing a high-performance public blockchain, decentralized storage, and distributed threshold cryptography, Nexa aims to provide a secure, scalable, and cost-effective solution for EHR systems. This architecture seeks to transform healthcare blockchain from a theoretical concept into a viable prototype capable of evolving into a production-ready solution for modern EHR systems. This will enable healthcare providers to deliver uninterrupted care while simultaneously protecting patient data against persistent cyber threats.

1.2. Problem and Research Questions

The integration of blockchain in EHR systems presents a challenge in balancing scalability, security, and decentralization. While blockchain can strengthen data integrity and patient control, many of the existing systems suffer from high transaction costs (e.g., unoptimized smart contracts logic) [2], limited throughput (e.g., improper choice of blockchain), and increased risks of centralization (e.g., due to the reliance on permissioned blockchains) [3]. This thesis proposes an architecture that attempts to balance security and scalability. By integrating a unique cryptographic scheme with a scalable public blockchain, it ensures the system benefits from strong security,

 doi:10.20944/preprints202602.0280.v1

scalability, and decentralization to push blockchain-based EHR systems further towards large-scale healthcare adoption.

To achieve this, the thesis focuses on the following Research Questions (RQs):

- **Research Question 1:** "How does Nexa, through its selection of a public blockchain with specific characteristics, achieve an effective balance of the blockchain trilemma for secure and efficient EHR management?"
- **Research Question 2:** "How does Nexa's distributed threshold cryptographic scheme, with its efficient algorithms and oracle design, better secure EHRs and manage access without excessive overhead?"

The research framework will primarily be quantifiable and will address the research questions directly by ensuring this workflow is followed:

- Theoretical evaluation of different public blockchain and decentralized storage solutions to assess trade-offs based on the blockchain trilemma.
- Propose a distributed cryptographic mechanism for encrypting EHRs to maintain data security without compromising system scalability.
- Optimize both the data flow between system participants, as well as the smart contract logic, to minimize costs and computational overhead.
- Conduct benchmark testing using synthetic medical datasets that emulate real medical datasets to measure system performance (latency).

1.3. Scope and Limitations

Since this thesis is conducted at undergraduate level, it will remain within a narrow scope, concentrating primarily on proposing an initial architecture. Implementation may simulate environments rather than deploy production-level ones, serving strictly as a proof-of-concept. This approach leaves considerable room for future work to refine and tailor the architecture for specific use-cases.

The research methodology will focus on both: 1. theoretical comparative analysis to evaluate various technical solutions and 2. practical validation through the technical implementation of the proposed architecture in the comparative analysis, including its benchmark testing with synthetic medical data to ensure reliability and to obtain quantifiable values that translate into meaningful results. Real patient datasets, such as MIMIC or eICU, are excluded from tests due to ethical, legal, and privacy constraints, as well as institutional requirements like CITI program certification, which the authors do not possess. Therefore, synthetic data will be generated using the open-source tool Synthea [4] and will be utilized to closely emulate real medical data by producing FHIR-compatible EHRs. This approach overcomes potential practical obstacles, and it also ensures that simulations will accurately reflect some of the real-world operational conditions while minimizing the risks associated with managing sensitive medical data.

Moreover, the thesis acknowledges several other constraints due to its limited timeline. First, it is not possible to conduct practical experiments for all potential technical combinations discussed in the theoretical comparative analysis. Instead, the evaluation will focus on the one combination that aligns directly with the research criteria, based on the prioritization of key variables. Second, while the benchmark testing method provides valuable insights into scalability potential under load and the success of implementing the distributed cryptographic scheme, it does not entail all system components or extreme-scale scenarios. Third, qualitative insights from healthcare stakeholders like physicians and other healthcare professionals are omitted due to logistical constraints. Finally, legal compliance will not be explicitly addressed, but the architecture has the potential to adhere to it due to the use of powerful, approved cryptographic standards.

 doi:10.20944/preprints202602.0280.v1

2. Extended Background

2.1. Blockchain and the Blockchain Trilemma

Core Fundamentals

A blockchain is a decentralized ledger technology preserved by a network of nodes, where data is stored in cryptographically linked blocks. Transactions are initiated by users, validated via consensus mechanisms (e.g., Proof-of-Work [5] or Proof-of-Stake [6]), and added to an immutable chain. This structure eliminates reliance on centralized authorities, ensuring tamper-resistance and transparency [7]. Beyond financial applications, blockchain's decentralized nature makes it viable for managing sensitive records like EHRs, where data integrity and auditability are critical [8].

Blockchain Trilemma in Healthcare

According to the blockchain trilemma, three aspects could potentially limit a blockchain solution from achieving mass adoption if not balanced appropriately: 'Scalability,' 'Security,' and 'Decentralization.' The fundamental challenge lies in the fact that maximizing two of these aspects typically necessitates a significant compromise in the third, making simultaneous maximization of all three aspects practically infeasible [9]. However, at the time of writing, this remains a theoretical framework subject to potential future invalidation. This challenge explains the rapid growth of blockchain platforms, each attempting to solve the blockchain trilemma by maximizing these three critical aspects.

This trilemma has been largely overlooked in prior studies proposing blockchain-based EHR systems reviewed in the literature. This exclusion may stem from many such studies proposing permissioned blockchain implementations, which naturally compromise 'Decentralization' to enhance 'Scalability' and 'Security'. However, even studies proposing public blockchain solutions have not explicitly acknowledged this trilemma. Understanding the blockchain trilemma enables the design of systems specifically tailored for EHR applications. Compromises may be inevitable, but balance is important, which is why this thesis focuses exclusively on public blockchains, as permissioned blockchains inherently cannot balance the aspect of 'Decentralization' since it is compromised by their very nature [3] as stated earlier.

The healthcare industry requires both scalability and security due to the huge quantities of sensitive data involved, necessitating secure management of data access. However, one often overlooked aspect is 'Decentralization', which if implemented using the appropriate consensus mechanism impairs the feasibility of cyberattacks; this is because it makes breaches too costly to carry out for malicious actors [10]. Most of the public blockchain-based architectures proposed in the literature review prioritize security and decentralization through the reliance on Ethereum [11], as Ethereum is positively noted for these attributes, but less so for 'Scalability'. By addressing the balance gap and finding an optimal balance that also places importance on 'Scalability' with minimal trade-offs in 'Security' and 'Decentralization' (while still providing stronger security than current centralized systems), these EHR systems can be more feasibly adopted in production environments. In other words, any minimal trade-offs in 'Security' or 'Decentralization' would still offer substantially better security benefits compared to presently used centralized systems.

It is crucial to note that relying on a public blockchain does not mean that data must be publicly disclosed (non-encrypted). It simply means that anyone can participate in network governance by running their own nodes. EHRs will be encrypted using established cryptographic algorithms, with only non-sensitive metadata stored on-chain, an approach designed to comply with regulations such as HIPAA/GDPR.

 doi:10.20944/preprints202602.0280.v1

2.2. Decentralized Storage for EHRs

Decentralized storage enhances availability, as EHRs remain accessible even if some individual nodes go offline. Cost efficiency is another advantage, as decentralized storage reduces costs by up to 38% compared to traditional cloud solutions like AWS S3 [12]. However, write/read speeds are still a challenge, but could be mitigated through caching mechanisms in production environments.

3. Literature Review

Building upon the foundational understanding of blockchain technology and the blockchain trilemma, this literature review examines existing blockchain-based EHR proposals. Analysis of prior research reveals gaps and challenges that frame the motivation for an innovative architectural approach. The following review demonstrates how existing solutions have attempted to address the complex requirements of secure, scalable, and decentralized healthcare data management.

3.1. Review Approach

A comprehensive snowballing literature review was applied to investigate blockchain applications in EHR systems. This methodical approach ensured a wide and comprehensive exploration of existing research through a structured process. It covered these steps:

- **Keyword Search Strategy:** An iterative approach was utilized, applying multiple keyword combinations across various academic databases to ensure extensive coverage. Keywords such as:
  - “Blockchain” AND “Electronic Health Records”
  - “Distributed Ledger Technology” AND “Healthcare”
  - “Blockchain” AND “Medical Data Security”
  - “Decentralized” AND “Patient Records”
  - “Smart Contracts” AND “Healthcare”
- **Initial Seed Sources**: Targeted searches were conducted across multiple academic databases, including but not limited to:
  - PubMed
  - IEEE Xplore
  - ACM Digital Library
  - ResearchGate
  - Google Scholar
  - Science Direct
- Forward Snowballing
  - Analyzed citations in key research papers
  - Tracked newer research citing foundational works
  - Expanded research scope through cited references
- Backward Snowballing
  - Reviewed reference lists of initial sources
  - Explored historical development of blockchain in healthcare
- **Inclusion Criteria**: The study focused on sources meeting the following specifications:
  - Journal publications and technical whitepapers
  - English-language sources
  - Content specific to Blockchain and Cryptography

 doi:10.20944/preprints202602.0280.v1

- Published within the last 10 years (only for blockchain papers, accounting for the rapidly evolving blockchain field)
- Papers proposing blockchain-based EHR systems

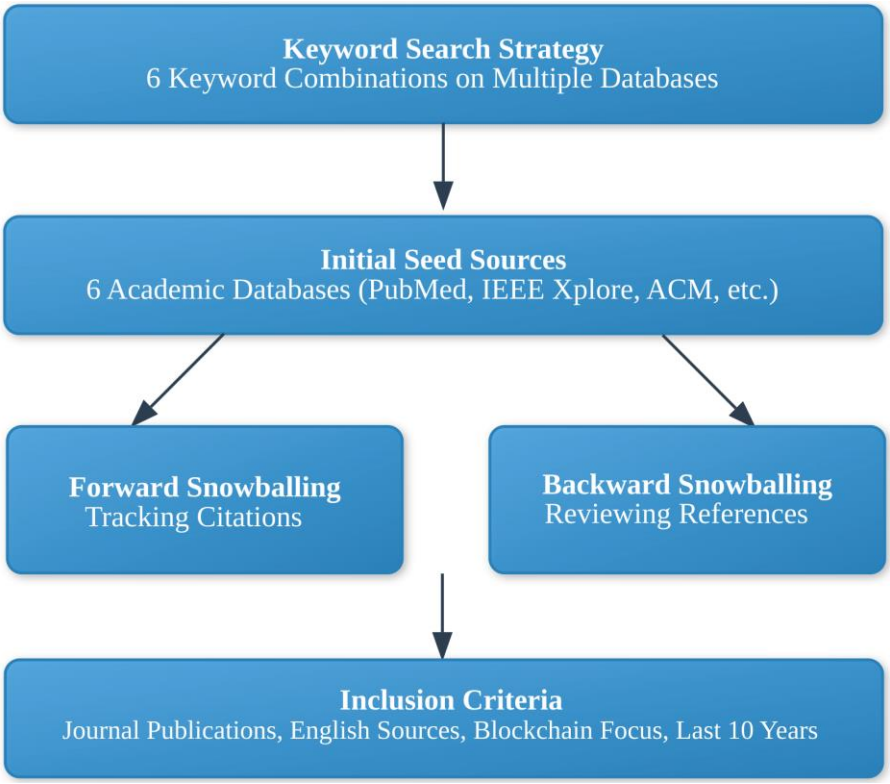


Figure 3.1. Diagram illustrating the selection process of studies included in the analysis.

3.2. Research Findings

The literature review revealed seven blockchain-based EHR proposals categorized into three primary architectures: public blockchain architecture, permissioned blockchain architecture, and hybrid architecture. Each architecture offers distinct strategies for addressing critical challenges in healthcare data management.

Public Blockchain Proposals

- **MedRec [13]**: One of the earliest proposals. An Ethereum-based system utilizing smart contracts to enable patient-centric medical history access. Despite being pioneering, the system encountered significant challenges, particularly regarding Ethereum's transaction throughput limitations.
- **HealthRec-Chain [14]:** Developed on an Ethereum fork and integrated with InterPlanetary File System (IPFS) [15], this proposal attempted to address blockchain scalability by storing large EHRs off-chain. However, the system faced challenges, with cost optimization being problematic due to the requirement of deploying a new smart contract for each patient-physician visit.
- **EdgeHR [16]**: A decentralized architecture proposing a novel approach of storing EHRs locally on user devices, with the Ethereum blockchain serving as a data indexing mechanism. The reliance on Ethereum and peer-to-peer LAN can limit scalability and increase vulnerability to network attacks like Man-in-the-middle (MITM) and eavesdropping.

 doi:10.20944/preprints202602.0280.v1

Permissioned Blockchain Proposals

- **ACTION-EHR [17]**: This system utilized Hyperledger Fabric [18], prioritizing regulatory compliance through network participation controls for authorized entities. Its centralization increases vulnerability to attacks.
- **MedChain [19]:** A peer-to-peer network designed for secure, immutable healthcare data management. The system ensures data integrity through digest chains and enhances scalability with session-based access control; it enables efficient and secure sharing of EHRs and IoT data. A higher risk of centralization has been identified.
- **Sec-Health [20]:** A blockchain-based protocol ensuring secure health record management by addressing regulatory requirements (confidentiality, access control, integrity, revocation, anonymity). It combines CP-ABE encryption for data stored on IPFS with blockchain-stored metadata for tamper-proof verification. Emergency access uses threshold cryptography, while revocation deletes transaction data. The use of a permissioned blockchain and minimal reliance on threshold cryptography increases the risk of centralization.

Hybrid Blockchain Proposals

- **SC2M-EHR-B [21]**: A proposal combining blockchain technology's security capabilities with the cloud's scalable storage infrastructure, representing an approach to addressing system scalability limitations. Centralized storage can make the system vulnerable to Denial-of-Service (DoS) attacks targeting stored EHRs to cripple availability.

3.3. Key Challenges Identified

Three primary challenges were concluded from the comprehensive review:

- **Blockchain Trilemma:** The fundamental challenge of simultaneously achieving scalability, security, and decentralization remains a critical constraint. Existing solutions often improve certain aspects at the expense of others, even if unintentionally.
- **Data Storage:** Efficiently managing large healthcare datasets in a decentralized manner without incurring excessive storage costs and latency continues to represent a significant technological hurdle.
- **Key Management:** Integrating strong, distributed key management schemes that prevent single points of failure while maintaining optimal accessibility and ease of use remains a critical research area.

3.4. Contribution of the Thesis

This thesis addresses existing gaps through original integration of several technologies:

a. Avalanche [22], a scalable public blockchain with 3,400+ TPS, 2-second finality, and subnet sharding, to overcome Ethereum's throughput limitations while maintaining decentralization.
b. IPFS for cost-efficient, tamper-proof EHR storage, coupled with on-chain content identifiers and integrity hashes.
c. Distributed threshold cryptography (the use of secret keys to protect EHRs, these keys are then divided into shares using a threshold splitting scheme, with each share thereafter encrypted and distributed via asymmetric encryption), serving as a replacement for centralized key management. By distributing the key shares across patients, physicians, healthcare institutions, and oracles, the system ensures no single entity can decrypt records alone, which helps in preventing breaches even if a few of the shares get compromised.

This approach directly responds to the trilemma: Avalanche's performance addresses scalability, distributed threshold cryptography enhances security without risks of centralization, and decentralized storage ensures data availability. By combining these elements, the architecture could offer a pragmatic solution for modern healthcare ecosystems. However, this work represents an

 doi:10.20944/preprints202602.0280.v1

initial step, and further research is needed to optimize performance, enhance interoperability with legacy systems, and validate the system in real-world healthcare environments.

4. Method

4.1. Research Strategy

This thesis follows a quantitative and design-oriented strategy aimed at evaluating technical feasibility and performance of a decentralized architecture for EHR management. The core objective is to introduce Nexa, a blockchain-based EHR architecture that attempts to balance scalability and security by assessing how modifications in blockchain platforms, data storage, and cryptographic schemes could affect the total system performance.

The thesis is structured around a comparative approach, in which key design components, specifically public blockchains, decentralized storage networks, and lightweight cryptographic algorithms are evaluated and compared individually. The methodology aligns with engineering research practices, focusing on analyzing system behavior under controlled, replicable conditions to assess trade-offs among cost, speed, and security.

A core part of the strategy involves evaluating a practical implementation. Key performance indicators (KPIs), such as transaction throughput, data storage and retrieval latency, and encryption/decryption overhead are measured. These metrics are evaluated using realistic, synthetic EHR datasets within a testing environment that closely simulates a production-level setting, ensuring controlled yet practical benchmarking conditions.

In essence, the strategy involves systematically comparing and evaluating individual system components to identify performance bottlenecks, optimize them, and eventually integrate these enhancements into a practical proof-of-concept implementation.

This strategy directly supports the research questions by making it possible to quantify how design choices in a blockchain-based system affects EHR management. By assessing multiple independent variables: blockchain platform, storage configuration, cryptographic algorithms, each layer's contribution to overall system performance can be isolated and interpreted.

4.2. Data Collection and Usage

Due to ethical and legal limitations on using real patient EHRs, this thesis utilizes publicly available synthetic EHR datasets. These datasets adhere to the Fast Healthcare Interoperability Resources (FHIR) standard, emulating realistic EHRs to enable accurate system performance evaluation without compromising sensitive patient data.

The data was generated using Synthea [4], an open-source tool that creates lifelike, diverse, and variable-sized synthetic patient EHRs. A total of 1,000 records were generated using Synthea to ensure statistical significance, resulting in approximately 4 GB of data. The average record size is around 4 MB, with the smallest record approximately 600 KB and the largest around 80 MB [23].

The dataset has the following characteristics:

- EHR records of varying sizes, ranging from approximately 600 KB to 80 MB, representing different types of text-based entries (e.g., clinical notes, diagnostic reports, imaging data). Each file represents a patient persona.
- Files in JSON format, compatible with the FHIR standard, which is commonly used in modern EHR systems.

This data was used in simulation experiments to measure:

- Public blockchain transaction latency when submitting EHR metadata.
- Storage and retrieval latency in decentralized storage systems.
- Cryptographic overhead when applying symmetric encryption to records.

By leveraging synthetic data that closely emulates real-world EHRs, this approach ensures both replicability and alignment with authentic data structures. Although real patient datasets were

 doi:10.20944/preprints202602.0280.v1

excluded due to ethical and legal considerations, the thoughtfully generated synthetic dataset, with a focus on variability and randomness, provides a strong foundation for the technical validation of system performance within the proposed EHR architecture.

4.3. Technology Selection Framework

Considered Variables for Blockchain Selection

- **Transaction throughput (TPS):** This refers to the number of transactions a blockchain can process concurrently per second. This metric primarily applies to write operations, as read operations are generally not classified as transactions and incur no cost on the chain; this is because they occur from a single node and do not require heavy computational power to execute. An EHR system must demonstrate reliability in managing high-volume data operations, as it is estimated that a single healthcare provider may store up to 80 MB per patient annually [24], which indicates a consistent pattern of frequent data writing and reading on the system.
- **Finality time (transaction latency):** This refers to the duration required for a blockchain to finalize a transaction, rendering it irreversible. This variable is critical, as certain blockchains, like Bitcoin [5], can take a considerable amount of time to finalize a transaction, depending on network congestion. A study analyzing nine years of Bitcoin transaction history, between the period from January 3rd, 2009, to April 30th, 2018, revealed that the majority of valid transactions were finalized within a timeframe ranging from 1 hour to 2 hours after receiving 6 confirmations [25]. For vital data, this duration can be extremely long. In the context of an EHR system, transactions must be finalized within seconds or in near-real-time to enable healthcare providers to verify data entry during patient visits. The primary reason for minimizing finality time is to quickly determine whether a transaction has been confirmed or remains unprocessed. If a transaction containing valuable references to medical data remains unconfirmed for too long, there is a risk that it may be dropped, delaying data availability. Therefore, minimizing finality time enables medical staff to receive success/failure responses from the system almost instantly, allowing them to resend the transaction if needed.
- **Fee structures:** This refers to the calculation of gas fees, which are required for executing transactions on the blockchain, typically involving write operations. This variable is crucial as it directly impacts the scalability potential of the EHR system. For instance, the Ethereum blockchain experienced steep gas fee levels, with a single simple transaction averaging $11.36 from April 14, 2020, to April 14, 2023 [26] (the period spans both before and after The Merge, meaning the pre-Merge average could have been higher). This surge was mostly driven by high demand and Ethereum's auction-based fee model, where users paid higher fees to prioritize their transactions. When selecting a blockchain for an EHR system, it is essential to consider fee structures and potential scenarios that could influence these fees.
- **Sharding Capabilities:** This refers to the ability of a certain blockchain to divide the network into smaller, parallel subchains (called "shards"), each responsible for processing a subset of transactions. This reduces the computational load on any single chain, helping to mitigate network congestion, which occurs when the blockchain is overwhelmed by a high volume of transactions [27]. For the EHR system, building it on a blockchain that has sharding capabilities or planning to implement them could significantly improve its potential to scale. This would enable the EHR system to remain optimized and efficiently manage transaction throughput, even during periods of blockchain congestion.
- **Block Time:** This refers to the average time interval between sequential blocks being added to the blockchain, which impacts transaction throughput and confirmation latency. Higher block time intervals may result in reduced decentralization because they typically correspond to larger blocks that demand more storage space and require more advanced hardware. These increased

 doi:10.20944/preprints202602.0280.v1

requirements can limit the ability of entities to host nodes, negatively impacting the network's topology and increasing the risk of centralization [28].

- **Consensus mechanisms:** This refers to the methods that blockchain networks use to validate transactions; this has a direct impact on how many transactions can be processed (TPS) and also on finality time (transaction latency). For example, Bitcoin uses Proof-of-Work (PoW), which ensures strong decentralization and security through open participation and mining. However, it struggles with scalability and sustainability because it requires a lot of computing power, which results in substantially slower transaction processing speeds and considerable energy consumption [29]. Other consensus models aim to address these issues, and the most time-tested ones are:

a. **Proof-of-Stake (PoS):** Validators are chosen based on how much funds they have staked (locked up) in the network. Higher stake quantities correlate with increased validator selection likelihood. PoS offers similar decentralization and security as PoW but is considerably more energy-efficient and scalable.

b. **Delegated Proof-of-Stake (DPoS) [30]:** Token holders vote for a smaller group of delegates to validate transactions. This improves scalability and maintains decentralization and security, though it comes with some risks of centralization since fewer validators are typically involved.

c. **Proof-of-Authority (PoA) [31]:** Transactions are validated by trusted pre-chosen entities with a good reputation. This mechanism adopts a centralized structure, prioritizing rapid transaction finality and operational efficiency at the expense of permissionless participation.

For EHR systems, both PoS and DPoS consensus algorithms are strong candidates, as they offer good scalability without the significant centralization risks associated with PoA. While the PoW consensus mechanism excels in terms of security and decentralization, it performs poorly in terms of scalability.

- **Smart Contract Access and Visibility Modifiers:** This refers to the ability of a blockchain to hold immutable smart contracts that enforce access control through explicitly defined roles. This functionality includes:

a. Access modifiers, which manage permissions and determine who can execute specific functions or logic within the smart contract.

b. Visibility modifiers, which help restrict external access to internal data by leveraging native privacy features. For example, using the *private* visibility modifier in Solidity [32] to prevent external retrieval of metadata (though it does not provide true confidentiality, only obfuscation through added complexity).

This thesis identifies the public blockchain that maximizes scalability for the proposed EHR architecture by balancing these key variables.

Considered Variables for Decentralized Storage Selection

- **Expandability:** The ability to handle increasing volumes of data and concurrent access over time. This includes factors such as the number of active storage nodes, the expandability of the available storage pool, and the protocol's ability to maintain performance under increasing load.
- **Storage Cost:** Estimated monthly cost per terabyte, including any token requirements or collateral models.
- **Data Immutability:** The ability for data to remain permanently accessible poses a potential security risk, particularly as cryptographic algorithms evolve. For instance, the advent of quantum computing could render current cryptographic techniques obsolete, which compromises the integrity of immutable data.
- **Degree of Decentralization:** Architectural characteristics reflecting control distribution, including reliance on federated or centralized nodes.
- **Integration Feasibility:** The presence of extensive SDKs, detailed API documentation, efficient mechanisms for real-time read/write operations.

 doi:10.20944/preprints202602.0280.v1

4.4. Reliability and Validity

Ensuring the reliability and validity of this research was essential for evaluating the scalability and performance of the proposed blockchain-based EHR system. The study followed a structured and replicable methodology, using controlled simulations, standardized synthetic datasets, and open-source technologies to assess system behavior under various configurations.

Reliability was achieved by maintaining consistency across experimental runs. All components including encryption algorithms, blockchain environments, and decentralized storage systems, were tested under identical conditions. The codebase is modular, which enables isolated testing of each layer (blockchain logic, storage layer, encryption mechanism).

Internal validity was maintained by aligning the metrics used in the experiments directly with the research questions. Metrics such as transaction throughput, latency, encryption overhead, and storage cost were chosen because they represent clinical dimensions of system performance in a real-world EHR context. Each test was designed to isolate a specific variable (e.g., blockchain type, storage backend, encryption method) so that causal effects could be observed and interpreted with minimal interference from unrelated factors.

External validity, or generalizability, was addressed by using synthetic healthcare datasets that closely mimic real-world EHR records in both structure and content. These datasets followed FHIR standards and included a variety of data types (e.g., text, diagnostic codes, imaging metadata). While the system was not deployed in an actual healthcare institution, the experiments were run on public blockchain testnets and real-world decentralized storage networks, reflecting environments similar to those in which a future production system would operate.

Despite these efforts, certain limitations remain. The use of synthetic data prevents assessment of how the system performs in actual clinical workflow involving real patients and healthcare providers. Furthermore, the system's user-facing aspects, such as interface usability, onboarding, or clinical workflows, were not evaluated, as the study focused on architectural and technical layers. These aspects would require further validation through qualitative studies or pilot deployments.

Nonetheless, within the technical scope of this thesis, the research design ensures that the results are reliable, valid, and applicable to the evaluation of scalable, secure, and efficient blockchain-based EHR architectures.

4.5. Ethical Aspects

The ethical considerations in this study primarily concern the protection of sensitive health data, compliance with legal standards, and responsible system design for future deployment in real-world healthcare settings. This research was conducted without human participants and utilized only synthetic data; therefore, formal Institutional Review Board (IRB) approval was not required. However, the study design adheres to ethical principles that would guide future implementation involving real patients.

First, the decision to use synthetic data exclusively was made to comply with privacy laws such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). These regulations restrict the use of real patient data without proper consent or safeguards, even in research settings. By using synthetically generated datasets from tools like Synthea, this research minimizes privacy risks while maintaining realistic data structure and diversity.

Second, the system design ensures that data ownership and access control remain with the patient. The use of blockchain-based identity verification and smart contract permissions aligns with the principle of patient autonomy, allowing only authorized healthcare providers to access medical records.

Finally, although no human participants were involved in this study, the research adheres to ethical standards relevant to digital health system development. If deployed in a real-world setting, the system would undergo further ethical review, particularly concerning user experience, consent mechanisms, and responsibilities related to data custody.

 doi:10.20944/preprints202602.0280.v1

5. Architectural Design and Implementation

5.1. Comparative Analysis

Decentralized Storage

To design a secure and privacy-compliant system for EHRs, a decentralized storage architecture was chosen to ensure availability through a distributed data model. In this model, data can be split into chunks, optionally encrypted, and stored across a network of independent storage nodes. This approach reduces the risk of a single point of failure and DoS attacks, and limits the possibility of any centralized entity having full access to sensitive data, provided that robust encryption and access controls are in place.

To determine the most suitable decentralized storage for Nexa, four candidates (Filebase [33], Sia [34], Filecoin [35], and Arweave [36]) were selected based on variables discussed in the 'Method' section.

Public Blockchain

Selecting the correct public blockchain is crucial, as an improper one may cause transaction bottlenecks or high costs, negatively affecting usability of the system. This section of the analysis evaluates top public blockchains using quantitative variables: TPS, finality time, fees, sharding, block time, consensus mechanisms, and smart contract modifiers. When these factors are properly balanced, they enhance scalability, without heavily compromising the security and decentralization aspects of the chain.

Previous studies that proposed public blockchain-based architectures, as reviewed in the literature, have not thoroughly analyzed their choice of blockchain or addressed the blockchain trilemma in depth. This subsection aims to fill that gap by providing an analytical exploration and justification for selecting one public blockchain over another based on the variables in the 'Method' section.

Comparison of Public Blockchain Networks

Network	Consensus	TPS*	Finality Time	Fee / Transaction	Sharding	Modifiers in Contracts	Block Time
Bitcoin	PoW	7	~60-120 min	~$1-5	No	No modifiers supported	10 min
Ethereum	PoS	15-30	15 min	~$0.5-10	Planned	Access and visibility modifiers	12 sec
BNB Chain	PoS	2,000+	3-6 sec	~$0.02-0.10	No	Access and visibility modifiers	3 sec
Solana	PoS	7,200+	400 ms	<$0.001	No	Only access modifiers	400 ms
Cardano	PoS	800+	20 sec	~$0.1-0.3	Planned	Only access modifiers	20 sec
TRON	DPoS	2,000+	3 sec	<$0.01	No	Access and visibility modifiers	3 sec
Avalanche	PoS	3,400+	2 sec	~$0.1-1	Yes	Access and visibility modifiers	2 sec
Polkadot	PoS	1,000+	6 sec	~$0.01-0.10	Yes	Only access modifiers	6 sec
NEAR	PoS	12,000+	2-3 sec	<$0.01	Yes	Only access modifiers	1 sec
Polygon	PoS	7,000+	3-5 sec	<$0.01	No	Access and visibility modifiers	2 sec
Arbitrum	Rollup**	4,500+	7 days***	~$0.1-0.5	No	Access and visibility modifiers	2-10 sec
Optimism	Rollup**	2,000+	7 days***	~$0.1-0.5	No	Access and visibility modifiers	2-10 sec

Figure 5.1. Comparison matrix of leading public blockchain networks as of May 2025. (*) TPS may vary over time; the values presented reflect the potential throughput each network is projected to achieve based on research and analysis as of May 2025. (**) An Optimistic Rollup is a Layer 2 solution that processes transactions off-chain, assumes they are valid, and relies on a challenge period to catch fraud. It

 doi:10.20944/preprints202602.0280.v1

reduces Ethereum congestion and acts as a subchain, similar to sharding but without changing Ethereum itself. (***) Optimistic Rollups have a 7-day challenge period for final settlement, but correctly signed and uncontested transactions are often considered practically final within minutes.

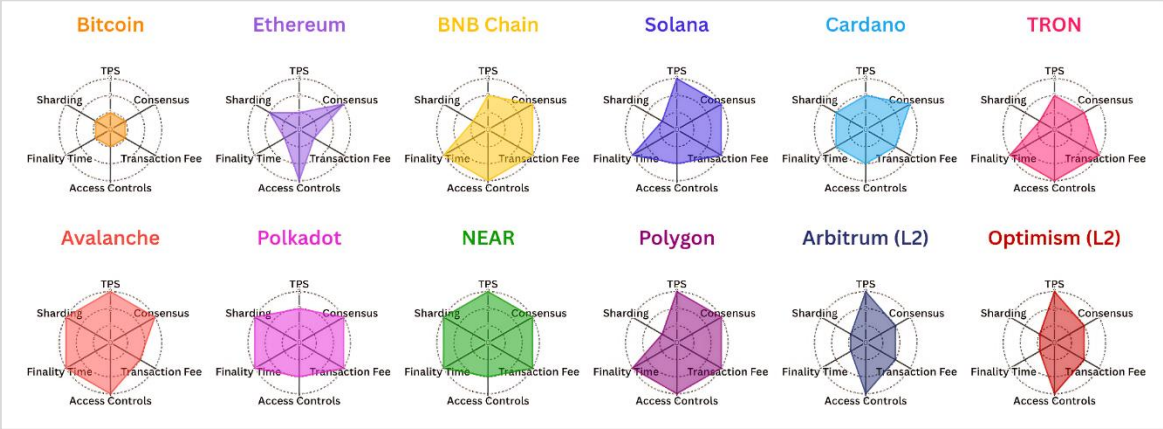


Figure 5.2. Radar chart comparing 12 blockchain networks based on criteria from Figure 5.1. Avalanche and NEAR display the most balanced profiles for EHR use-cases, while Bitcoin and Ethereum show notable limitations..

Distributed Threshold Cryptography

The literature review reveals that, although threshold cryptography has been employed in prior blockchain-based EHR systems (see Sec-Health [20]), it has not been explored as the primary cryptographic approach. As a result, there is limited understanding and insights into how it performs in a broad context. Most other existing proposals use centralized key management mechanisms, introducing a major weakness: if a patient's or physician's private key is compromised, the entire EHR could become exposed.

This thesis explores the use of threshold cryptography within a distributed framework to enable secure and efficient access to EHRs by system participants. The key idea is to use Shamir's Secret Sharing [37], which splits the symmetric encryption key into multiple parts and distributes them among decentralized participants across various locations and systems. This design ensures that even if an attacker compromises a single participant private key and obtains one share; they cannot reconstruct the full encryption key or access the EHR data, as multiple shares are needed for decryption.

The implementation utilizes the ChaCha20-Poly1305 stream cipher [38] to secure EHR data, as it is more efficient than AES in terms of encryption and decryption speed, particularly on systems without dedicated hardware acceleration [39]. Moreover, it allows for data integrity verification using the Poly1305 auth tag. During the creation of an EHR, a symmetric key is securely split into five shares $(n = 5)$ using Shamir's Secret Sharing, with a threshold of at least three key shares $(k \geq 3)$ required for reconstruction. This process occurs on a secure device operated by a physician within the healthcare provider's environment (e.g., a hospital). This approach ensures that no single participant can reconstruct the entire key on their own, while still enabling efficient access for authorized users. Although higher values of n and k can further strengthen security, this thesis adopts the $(n = 5\ ,\ k \geq 3)$ configuration to reduce both cryptographic computation and communication overhead among system participants.

For each share of the symmetric key, an asymmetric encryption scheme is applied using the public key of the respective system participant (in this case: the patient, hospital, physician, and two neutral decentralized oracle services) and an ephemeral key pair. This approach ensures that the original encryption key can only be reconstructed when multiple authorized entities work together, reducing risks from compromised credentials. Elliptic Curve Diffie-Hellman (ECDH X25519) [40] is

 doi:10.20944/preprints202602.0280.v1

selected for its efficiency, offering comparable or even slightly better security than RSA due to shorter key lengths which results in reduced computational overhead and improved performance [41].

Oracle services act as neutral third-party validators deployed by hospitals (or other trusted entities) in this architecture. They store their private keys securely and only handle decryption requests of any ChaCha20-Poly1305 shares after validating that the requestor is authorized through the smart contract. The shares are re-encrypted using the requestor's public key before transmission, preventing interception during the exchange process.

The distributed threshold cryptography mechanism strengthens the system against attacks. Even if an attacker extracts encrypted shares from the blockchain, they would need to compromise at least three different ECDH-based private keys to reach the threshold for key reconstruction. This increases both difficulty and cost of attacks, making most breach attempts impractical. Additionally, this approach enables access revocation without re-encryption of existing EHRs. When access needs to be revoked, the smart contract logic blocks oracle services from delivering shares to revoked participants through a boolean check.

	Traditional Cryptography in EHR Systems	Threshold Cryptography (Proposed)
Key Management	Centralized (single key per user)	Distributed (multiple key shares)
Points of Failure	Single (compromise of one private key exposes data)	Multiple (requires $k \geq 3$ compromised private keys)
Revocation Mechanism	Requires re-encryption of records	Simple smart contract restriction
Trust Model	Concentrated (relies on individual users)	Distributed (across multiple entities)
Breach Impact	Complete data exposure	Limited (partial key shares only)
Complexity	Lower	Moderate to higher
Recovery Options	Limited (dependent on single key backup)	Flexible (can reconstruct from different combinations)

Figure 5.3. Comparison matrix of traditional cryptography and threshold cryptography.

5.2. Prototype Implementation

Machine Specifications and Development Tools

Hardware

- **CPU:** 13th Gen Intel i5-13500 (20 threads) at 4.800GHz
- **Memory:** 31852.29 MiB (32 GB RAM)
- **Operating System:** openSUSE Tumbleweed, 64-bit

These specifications are adequate for handling tasks like synthetic medical data generation, cryptographic operations, and blockchain interactions (for testing).

Development Tools

- **Languages:** TypeScript [42], Solidity
- TypeScript will be used for frontend development and smart contract integration, while Solidity will be used for developing the smart contracts.
- **Smart Contract Development:** Hardhat [43]
- Hardhat is utilized to develop, test, and deploy smart contracts on EVM-compatible blockchains.
- **Blockchain Integration:** Ethers.js, Viem

Ethers.js and Viem are used on both the frontend and in benchmarking scripts to interact with the blockchain via the JSON-RPC protocol, handling smart contract interactions and transaction management.

- **Storage:** IPFS (through Filebase as gateway). Provides decentralized storage solutions for encrypted EHRs.

 doi:10.20944/preprints202602.0280.v1

- **Frontend Framework:** React (with Vite as build tool)
- **Cryptography:** npm packages include *@noble/curves*, *@noble/ciphers*, and *shamir-secret-sharing*. Specifically, *@noble/curves* handles ECDH key pair operations. *@noble/ciphers* creates the ChaCha20-Poly1305 cipher and verifies the integrity of the EHR data. *shamir-secret-sharing* is used to split the ChaCha20-Poly1305 key into multiple shares.

System Architecture and Interaction Design

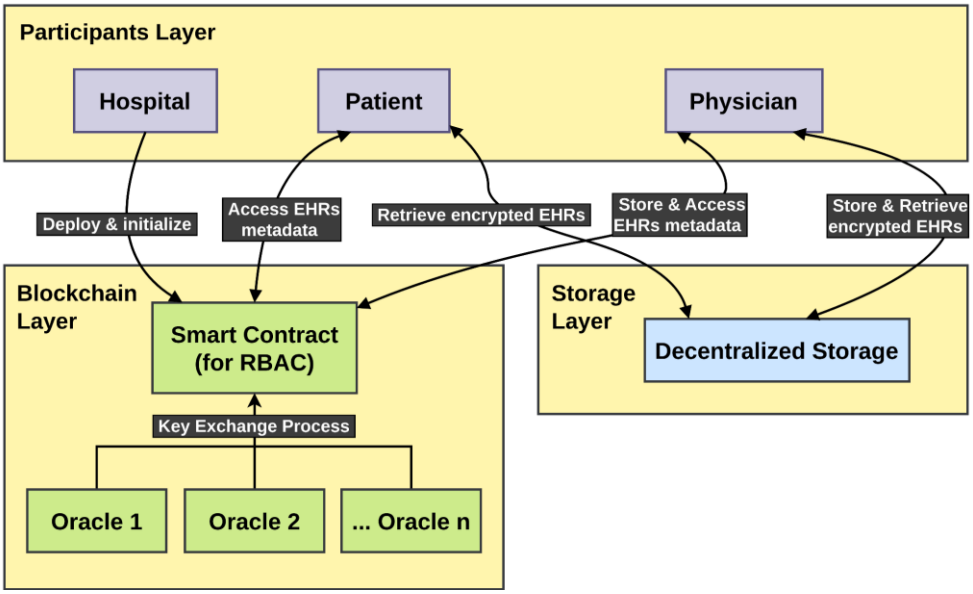


Figure 5.4. *High-level architecture diagram of the proposed EHR system, the arrows illustrate the data flow and interactions between components.*

A proof-of-concept system was built [44]. It consists of three main components:

1. **Participants Layer:** A React frontend interface that houses all system participants: Hospital administrators who deploy and initialize the system, Patients who can access and retrieve their own EHRs, Physicians who can store, access, and manage patient EHR data.
2. **Blockchain Layer:** This layer implements a smart contract that manages permissions for each participant using RBACs. It also facilitates a key exchange process that connects to off-chain oracle services. Acting as secure, neutral intermediaries, these oracles assist authorized requestors in reconstructing a ChaCha20-Poly1305 key, which was originally used to encrypt a specific EHR. The key itself is split using Shamir's Secret Sharing, ensuring the EHR remains protected even if a participant's private key is compromised.
3. **Storage Layer:** Consists of decentralized storage where the actual encrypted EHRs are stored off-chain, separate from the blockchain itself.

It is worth noting that all smart contract interactions shown in Figure 5.5, covering the Hospital, Physician, Patient, and Oracles, require a valid wallet address to authenticate and authorize communication between the user interface and the smart contract on the blockchain. For this thesis, it is assumed that each participant has access to their own MetaMask [45] wallet, which serves as the primary tool for interacting with the Solidity smart contract deployed on the Avalanche blockchain. In a production environment, MetaMask could be replaced by a custom-built wallet application tailored to meet specific hospital system requirements or adapted for particular regional regulatory and operational needs. These wallets utilize the Elliptic Curve Digital Signature Algorithm (ECDSA) over the secp256k1 curve, providing cryptographic mechanisms for securely authenticating and authorizing users' blockchain transactions. Importantly, these ECDSA-based wallets operate independently from the X25519 key pairs employed in Nexa's architecture for EHR encryption and

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 February 2026 doi:10.20944/preprints202602.0280.v1

decryption, ensuring a clear separation of concerns between blockchain identity/authentication and the cryptographic protection of sensitive medical data.

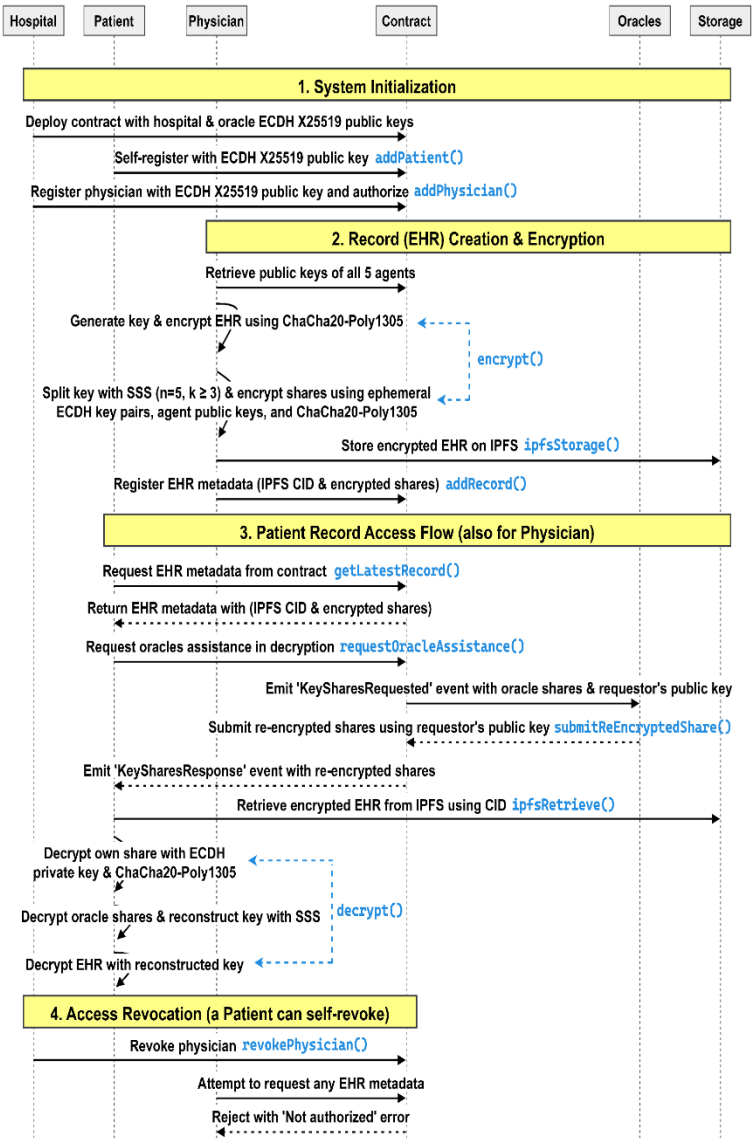


Figure 5.5. Sequence diagram showing actor interactions in the proposed EHR architecture. Blue-highlighted methods represent benchmarked operations discussed in the 'Results' section.

1. System Initialization Flow

The process begins with the hospital generating its own X25519 key pairs and securely storing the private key in a highly-protected offline vault to prevent unauthorized access. The hospital then deploys the Nexa smart contract on the blockchain, including its public key and the public keys of at least two oracle services, which will act as decentralized validators. Each oracle generates its own X25519 key pairs and securely stores its private keys in offline trusted environments, functioning as third-party neutral validators within the system.

Physicians also generate their own X25519 key pairs and store their private keys securely. They cannot self-register; instead, the hospital registers them in the smart contract by submitting their public key. This registration occurs only after the hospital verifies the physician's credentials offline, granting them access to patient records and the ability to request oracle assistance in decryption.

Patients generate their own X25519 key pairs and securely store their private keys in offline, protected environments. They can self-register their identities with the hospital's smart contract by submitting their public keys along with other non-sensitive personal data.

 doi:10.20944/preprints202602.0280.v1

2. Secure Record Creation Flow (Automated using React interface)

When creating an EHR during a patient-physician visit, the physician first retrieves all necessary public keys (patient, hospital, physician, and both oracles) from the smart contract. The physician then generates a ChaCha20-Poly1305 symmetric encryption key and nonce to secure the EHR data. After encrypting the record with this ChaCha20-Poly1305 key and generating a Poly1305 authentication tag, the physician applies Shamir's Secret Sharing scheme to split the ChaCha20-Poly1305 encryption key into five shares $(n = 5)$, with a threshold of at least three shares $(k \geq 3)$ required to reconstruct the original key.

For each share, the physician generates a new ephemeral X25519 key pair and derives a shared secret using the corresponding participant's public key. Each share is then encrypted with ChaCha20-Poly1305 using its derived secret. For example, the physician generates an ephemeral key pair for Share 1, derives a shared secret with the patient's public key, and encrypts Share 1; then repeats this process with a new ephemeral key pair and the hospital's public key for Share 2, and similarly for Share 3 (the physician), Share 4 (Oracle1), and Share 5 (Oracle2). The encrypted record is stored on decentralized storage (IPFS), which returns a content identifier (CID). Finally, the physician registers the record's CID and all encrypted key shares on the smart contract via the *addRecord()* function.

A comprehensive visualization and step-by-step explanation of this flow, including mathematical annotations, are provided in Appendix A.

3. Record Access Flow (Automated using React interface)

When a participant (patient or physician) needs to access an EHR, they request the record via the *getRecord()* or *getLatestRecord()* functions and authenticate with the smart contract using their MetaMask wallet. Upon successful authorization, the contract returns the EHR metadata, including the five encrypted shares and the content identifier (CID) for the decentralized storage. The participant decrypts their share using their private key and the shared secret derived from the ephemeral public key of the encrypted share.

To obtain the additional two shares needed to meet the threshold, the participant calls the *requestOracleAssistance()* function, which triggers the smart contract to emit a *KeySharesRequested* event, containing the encrypted oracle shares and the participant's public key. The oracles monitor for such events, decrypt their shares using their private keys and the shared secret derived from the ephemeral public keys, and then re-encrypt the shares using ChaCha20-Poly1305 with a shared secret derived from the participant's public key. The re-encrypted shares are then submitted back to the contract via the *submitReEncryptedShare()* function. The contract thereafter emits a *KeySharesResponse* event containing the re-encrypted shares.

The participant receives the re-encrypted shares, decrypts them, and combines them with their own share to reconstruct the entire ChaCha20-Poly1305 key. Using the reconstructed key, the participant retrieves the encrypted EHR from decentralized storage via the CID, decrypts it with the reconstructed key and the stored nonce, and verifies the Poly1305 tag to ensure the integrity of the contents.

A comprehensive visualization and step-by-step explanation of this flow, including mathematical annotations, are provided in Appendix B.

5. Access Revocation Mechanism

If a physician leaves the hospital or should no longer have access to patient records, the hospital can revoke their authorization in the smart contract using the *revokePhysician()* function. When the physician later attempts to request record access, the smart contract will verify their authorization status and reject the request with a "Not authorized" error. This provides an efficient access control mechanism, ensuring no changes are required to the already distributed shares or encrypted records.

6. Defense Against Breach

A malicious actor may attempt to bypass visibility modifiers and extract EHR metadata (i.e., encrypted shares, content identifier) from the smart contract. However, this does not constitute a vulnerability, as on-chain data confidentiality cannot be fully enforced through smart contract access control. Public blockchains are inherently transparent, and any metadata stored on-chain should be

 doi:10.20944/preprints202602.0280.v1

considered non-sensitive. Access controls are specifically designed to restrict the execution of logic based on defined roles.

To compromise a record, an attacker would need to break at least three private keys from different recipients to decrypt the required minimum number of shares $(k \geq 3)$. This means that even if an attacker compromises one share of the key, they would still need to break the encryption for each additional share, making the attack significantly more difficult. As a result, any attempt to breach the system becomes computationally infeasible due to the distributed encryption architecture.

6. Results

6.1. Theoretical Comparative Analysis

Selected Blockchain and Limitations

Based on the collected data and evaluated criteria in the 'Method' section, Avalanche emerges as the most suitable choice for implementing the proposed architecture due to the following advantages:

- **Data Integrity:** Avalanche offers superior data protection with both access and visibility modifiers in its Solidity-based smart contracts; this enables granular access control critical for sensitive EHRs.
- **Finality Time:** With 2-second finality, Avalanche provides almost instant transaction confirmation essential for physicians during patient visits.
- **TPS:** At 3,400+ TPS, Avalanche delivers powerful throughput easily capable of handling EHR system demands.
- **Sharding Capabilities:** Avalanche's subnet architecture provides native sharding functionality; this guarantees long-term scalability for growing EHR systems as subnets increase in number.

Limitations of Alternative Blockchains

In the comparative analysis presented in the 'Method' section, several alternative blockchains offer competitive performance but were not selected due to some specific limitations:

- **NEAR [46]:** Despite its excellent TPS (12,000+) and low fees, NEAR has a smaller ecosystem and fewer validators compared to Avalanche. EVM compatibility also provides better bridge functionality to other EVM chains, making Avalanche more practical for interoperability. NEAR remains a strong candidate worth investigating for future EHR systems.
- **Solana [47]:** While also offering high TPS (7,200+) and low fees, Solana has experienced multiple network outages https://status.solana.com/incidents/m6qzbgc7np9b, https://status.solana.com/incidents/ymr0gyj9xqy, https://status.solana.com/incidents/n5kcgs8dl9pj, posing substantial risks for critical EHR systems that require consistent availability. Additionally, the absence of visibility modifiers on Solana may impose further limitations on implementing extra privacy controls within smart contracts.
- **Polkadot [48] & Cardano [49]:** Their lower TPS (800-1,000+) compared to other high-performance networks makes them less appealing for the adoption of high-volume transaction demands of EHR systems.
- **BNB Chain [50] & TRON [51]:** Both offer low fees and fast transactions but lack sharding capabilities; this limits their long-term scalability for extensive healthcare data operations. BNB Chain utilizes Proof of Authority (PoA) as its consensus protocol, which raises concerns about decentralization, making it even less suitable for adoption in the proposed architecture.
- **Ethereum and Layer 2 Solutions:** These networks offer a mature ecosystem, but Ethereum suffers from higher fees and scalability issues. Layer 2 solutions like Arbitrum [52] and

 doi:10.20944/preprints202602.0280.v1

Optimism [53] have a delayed practical finality of ~10-15 minutes after a transaction is submitted and they could be deemed intolerable for time-sensitive EHR systems.

Limitations of Avalanche

Avalanche comes with some potential, non-critical trade-offs to consider:

- **Transaction Costs:** Avalanche's fees are higher than those of alternatives like NEAR or Solana but remain orders of magnitude lower than Ethereum's, which makes Avalanche reasonable enough to support adoption in an EHR system architecture when it comes to cost-efficiency.
- **Transaction Throughput:** Considering a transaction throughput of over 3,400 TPS, Avalanche could provide sufficient performance for current healthcare needs but as the comparison demonstrated, it falls below the capacities of NEAR (12,000+ TPS) and Solana (7,200+ TPS). However, it is important to note that Avalanche's subnet architecture and sharding capabilities can help mitigate this limitation as the network scales, making it a temporary constraint rather than a long-term barrier.

Selected Decentralized Storage and Limitations

All reviewed platforms offered decentralized file storage capabilities; however, their suitability for integration in healthcare environments varied considerably:

- **Sia:** Demonstrated compelling theoretical advantages, particularly in terms of cost-efficiency and native security. With built-in redundancy and decentralized peer-to-peer contracts, it maintains a highly competitive storage cost of approximately $1–$2/TB/month. However, its practical integration capabilities remain limited. At the time of evaluation, Sia lacked stable, well-maintained SDKs, provided minimal API documentation, and offered sparse integration support. These constraints greatly hindered its viability for automated deployments and prototyping, leading to its exclusion from the benchmarking phase despite its strong technical foundation.
- **Filecoin:** Built on top of the InterPlanetary File System (IPFS), introduced a robust framework for verifiable decentralized storage. It features cryptographic consensus mechanisms, Proof-of-Replication (PoRep) and Proof-of-Spacetime (PoSt), to ensure that storage providers reliably commit and retain data. While Filecoin benefits from a maturing ecosystem and ongoing adoption in archival and scientific contexts, its reliance on an auction-based storage market introduces variable upload latency and non-deterministic availability. Data uploads must pass through a bidding process before being committed to the network, complicating real-time interactions. Although well-suited for long-term archival use, this asynchronous mechanism made Filecoin less practical for the more interactive demands of EHR systems within the scope of this thesis.
- **Arweave:** Offers a unique value proposition in the form of permanent, one-time-payment storage. This immutability ensures that uploaded data is persistently accessible and tamper-proof. However, the inability to alter or delete content also introduces a vital long-term risk, especially in the context of sensitive health records. If current encryption algorithms become obsolete by future advances such as quantum computing, immutable datasets may become vulnerable, with no way to retract or sanitize previously uploaded content. For this reason, Arweave was deemed unsuitable for handling sensitive healthcare data where regulatory flexibility and the potential for future cryptographic upgrades must be considered.
- **Filebase:** Utilizes the IPFS protocol as its underlying storage mechanism, offering a familiar interface, well-documented SDKs, and up to 5 GiB of free storage for testing purposes. These features provided a strong foundation for research-stage integration. However, Filebase operates as a centralized gateway to the IPFS network, meaning that the party operating the gateway (the healthcare provider) maintains control over data retention, including the ability to delete records. Even though it is acceptable for prototyping and evaluation, this centralization

 doi:10.20944/preprints202602.0280.v1

introduces a potential single point of failure and must be addressed in any production-grade deployment. To mitigate this limitation in real-world scenarios, alternative solutions leveraging IPFS in a more decentralized way (without a centralized layer of control) are recommended.

Filebase was selected primarily for its cost-effectiveness (based on IPFS) and developer-friendly experience, which makes it suitable for controlled research purposes, despite its partial centralization. For production use, more decentralized IPFS-based alternatives (such as Web3.Storage or Storj) are generally recommended. These alternatives are not discussed here due to time constraints and setup complexity. However, it is important to note that since both Filebase and these other comparable solutions leverage the same underlying infrastructure, there will be minimal difference in terms of practical evaluations and results.

6.2. Benchmark Performance Analysis

Benchmarks [54] were conducted on the Avalanche Fuji Testnet using a dataset of synthetic FHIR-compliant EHRs. Time-sensitive operations, critical for real-time interactions, were evaluated using 1,000 transactions. Non-time-critical operations, such as administrative tasks, were tested with 100 transactions. The performance, in terms of latency and gas consumption for smart contract operations, is summarized in Figure 6.1 and Figure 6.2, respectively.

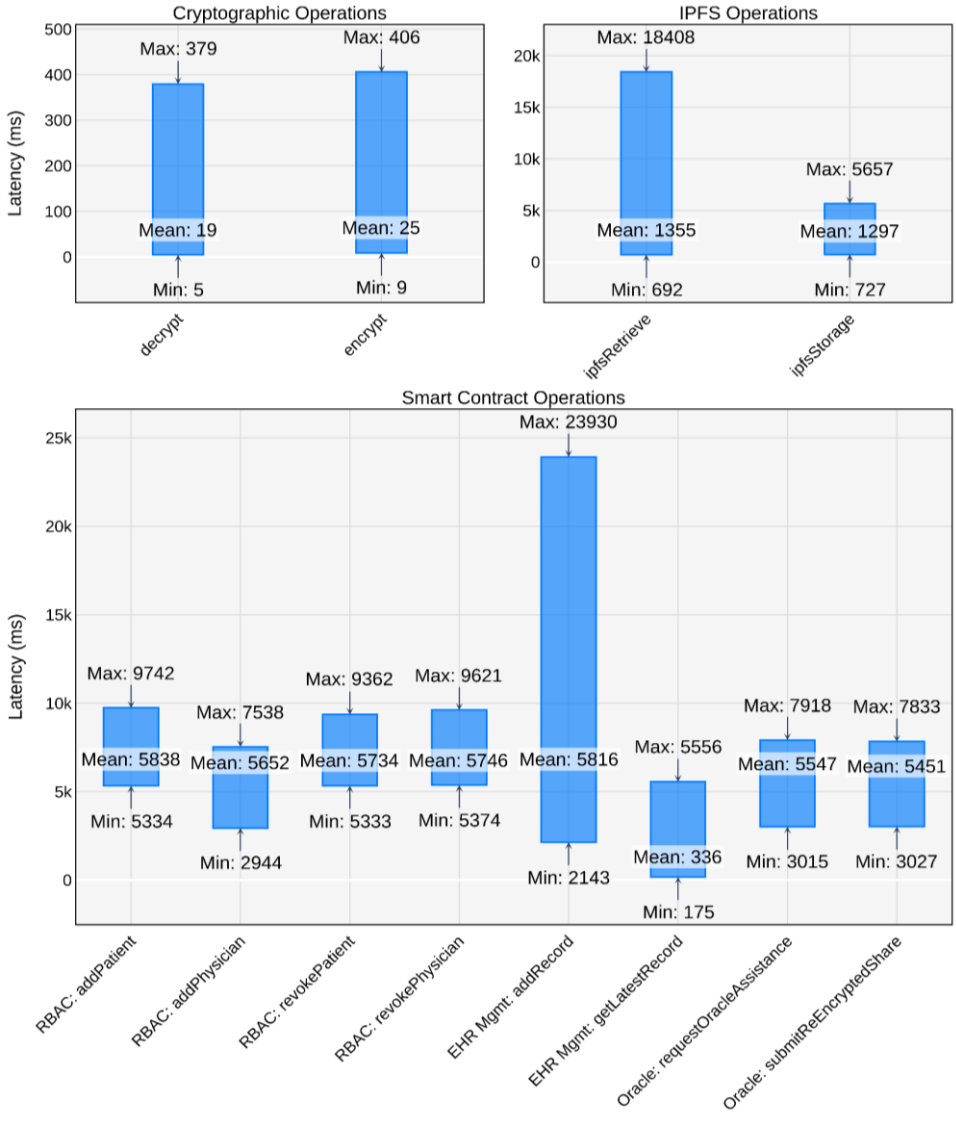


Figure 6.1. Latency (in milliseconds) across Smart Contract, IPFS, and Cryptographic operations.

 doi:10.20944/preprints202602.0280.v1

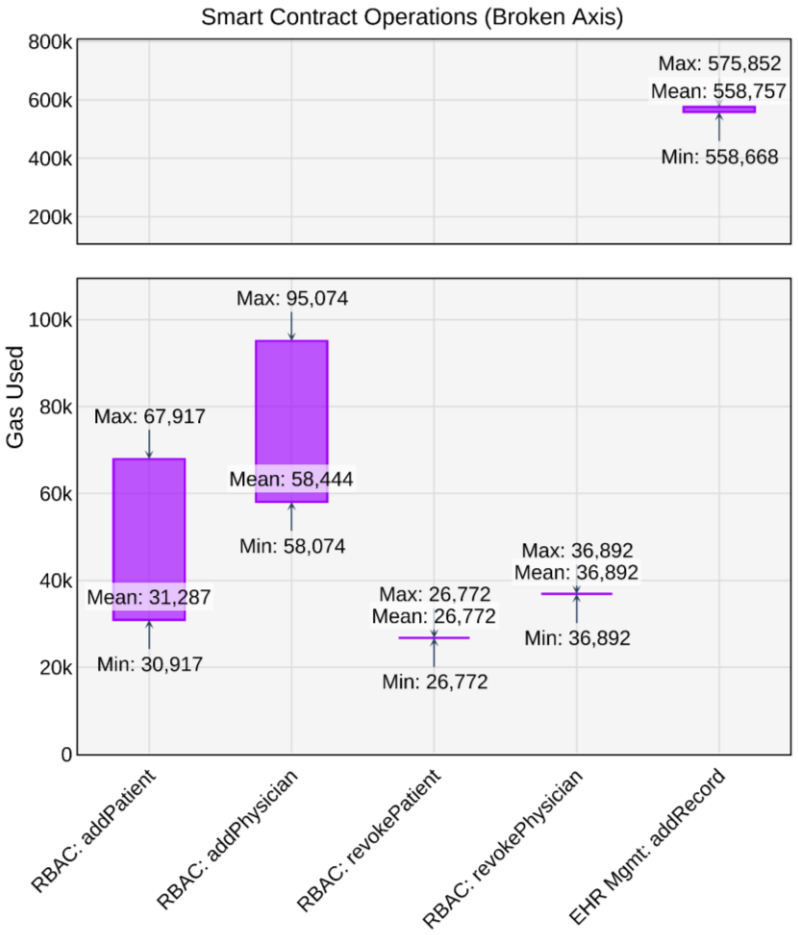


Figure 6.2. Gas usage (in units) for key Smart Contract operations on the Avalanche Fuji Testnet.

Smart Contract Operations

The smart contract primarily contains operations for RBACs, EHR metadata management, and oracle-assisted decryption. These operations were benchmarked to assess their viability. RBAC operations were tested with 100 transactions each, while EHR metadata operations (*addRecord()*, *getLatestRecord()*) and oracle interactions were tested with 1,000 transactions each, with the synthetic EHR dataset detailed in the "Data Collection and Usage" section.

RBACs and EHR Metadata Management

Operations related to role management (RBACs), including registering and revoking access for patients and physicians, are essential for system administration. As shown in Figure 6.1, these operations: *addPatient()*, *addPhysician()*, *revokePatient()*, and *revokePhysician()* demonstrate consistent mean latencies ranging from approximately 5,652 ms (for *addPhysician()*) to 5,838 ms (for *addPatient()*). The maximum observed latencies reached up to 9,742 ms (for *addPatient()*). These latencies suggest that onboarding a new user or modifying access rights would typically complete within 6-10 seconds.

In terms of gas consumption (Figure 6.2), RBAC operations are very cost-efficient. The *addPhysician()* operation exhibited the highest mean usage among them at 58,444 gas units, with a maximum of 95,074 units. Other RBAC operations like *revokePatient()* and *revokePhysician()* showed consistent gas usage around 26,772 and 36,892 units respectively. As of May 2025, with a high-priority gas price around 5 nAVAX (5×10^{-9} AVAX), the estimated transaction fee for the most gas-intensive RBAC operation (*addPhysician()*, at 95,074 units) would be approximately 0.000475 AVAX (roughly $0.011 USD, assuming an exchange rate of $24 per AVAX as of May 2025). The gas fee is calculated as:

$$Gas\ fee\ (AVAX)\ =\ Gas\ units\ \times\ (Gas\ price\ (nAVAX)\ \div\ 1e9)$$

 doi:10.20944/preprints202602.0280.v1

Critical to near-real-time clinical interaction is the management of EHR metadata. The *addRecord()* operation, for storing metadata on the blockchain, showed a mean latency of 5,816 ms (Figure 6.1). However, a maximum latency of 23,930 ms was observed. This maximum is considered an extreme outlier, likely due to temporary network congestion. Interquartile range (IQR) analysis (where IQR = 180, Q1 = 5,561 ms, Q3 = 5,742 ms) indicates an upper outlier threshold of 6,012 ms. Therefore, typical latency for *addRecord()* is expected to be between 5,291 ms and 6,012 ms, which could be within a feasible range for clinical workflows.

The *addRecord()* operation is the most gas-intensive, consuming a mean of 558,757 gas units (Figure 6.2), with a maximum of 575,852 units. This is substantially higher than RBAC operations due to storing more data (up to 653 bytes of metadata per single EHR). Despite this, the cost remains relatively modest: approximately 0.00279 AVAX (around $0.059 USD) per transaction on average, using the previously mentioned gas fee formula and corresponding price.

The *getLatestRecord()* operation, for retrieving EHR metadata, is highly efficient with a mean latency of 336 ms (Figure 6.1). Its maximum observed latency of 5,556 ms is also an outlier, given the IQR-based typical range of 208 ms to 432 ms (where IQR = 56 ms, Q1 = 292 ms, Q3 = 348 ms). Under normal conditions, this latency range is expected to be well within the parameters for near-real-time use. As a read operation, *getLatestRecord()* typically incurs no gas fees on the blockchain and is therefore not detailed in Figure 6.2.

Oracle-Assisted Operations

Operations involving oracle decryption assistance, *requestOracleAssistance()* and *submitReEncryptedShares()*, are integral to EHR retrieval and mainly emit smart contract events. Mean latencies for both operations were recorded at 5,547 ms and 5,451 ms, respectively (Figure 6.1). These latencies are influenced by Avalanche's Fuji transaction finality time, since emitting events requires logging to the blockchain, directly affecting the overall EHR retrieval latency. Although their gas consumption is not shown in Figure 6.2, these operations do not store state data on-chain and are therefore expected to be cost-efficient, with gas usage comparable to or lower than RBAC operations. On mainnet, the performance difference in total EHR retrieval latency compared to other architectures is expected to be minimal, and the trade-off of using smart contract–based events is justified by the system's enhanced security and decentralization.

Decentralized Storage Operations

Filebase was utilized to manage and pin files on the IPFS network. For this proof-of-concept prototype, Filebase's free-tier limits were used, incurring no storage costs. Latency benchmarks for storing and retrieving records from IPFS with *ipfsStorage()* and *ipfsRetrieve()* were measured using 1,000 synthetic EHRs to ensure content and file size variability (Figure 6.1, IPFS Operations subplot).

The results show promising latency: storing a record averaged 1,297 ms (max 5,657 ms), while retrieval averaged 1,355 ms (max 18,408 ms). With an average record size of around 4 MB, these findings suggest IPFS can handle EHR data efficiently. Outliers, reflecting realistic usage patterns, indicate that latency increases with record size. This highlights the importance of segmenting large EHRs into smaller records to maintain consistent performance.

Cryptographic Operations

The entire encryption and decryption processes were benchmarked. Encryption included key/nonce generation, ChaCha20-Poly1305 EHR encryption, Shamir's Secret Sharing (SSS) for key splitting, X25519 ephemeral ECDH key pair generation, shared secret derivation from ephemeral and participant keys, and ChaCha20-Poly1305 share encryption. Decryption mirrored these steps in reverse.

As shown in Figure 6.1 (Cryptographic Operations subplot), both operations are highly efficient. On average, the *encrypt()* operation took 25 ms (max 406 ms), and the *decrypt()* operation took 19 ms

 doi:10.20944/preprints202602.0280.v1

(max 379 ms). This efficiency stems from the selection of ChaCha20-Poly1305 and ECDH X25519 with their optimized key lengths. Maximum latencies were observed with very large files (≥ 80 MB), but performance remained competitive, potentially outperforming traditional algorithms like AES and RSA [39,41] for these use cases.

Comparison with other solutions

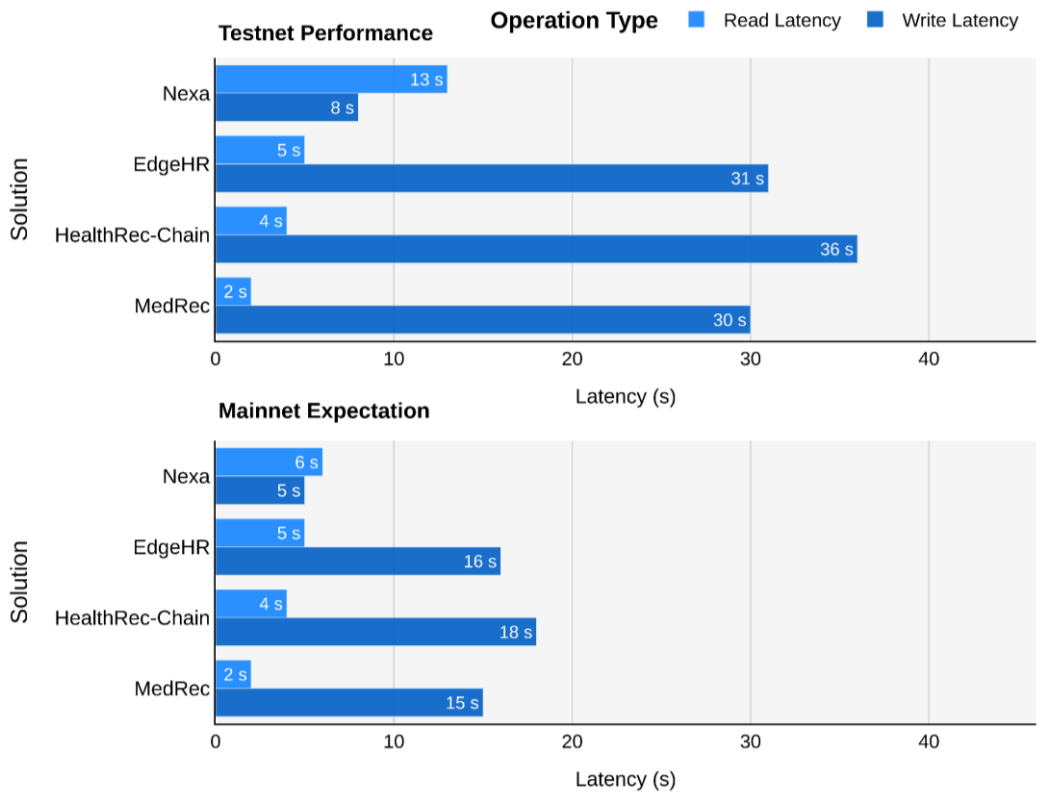


Figure 6.3. Comparison of total read and write latency between Nexa and other blockchain-based EHR solutions, showing both current testnet performance and projected mainnet expectations..

The comparison plot offers a detailed analysis of Nexa's total latency relative to other blockchain-based EHR solutions, showcasing both current testnet results and anticipated mainnet performance. Since not all competing systems have publicly available latency metrics, estimated values were used for platforms like EdgeHR and MedRec. These estimations were derived from the characteristics of their underlying technologies and architectural choices. For instance, EdgeHR utilizes P2P LAN communication alongside Ethereum as its blockchain foundation. Thus, its write/read latencies were estimated based on Ethereum's current performance benchmarks, plus the typical latency of EHR transfers over a P2P LAN connection.

On the testnet, Nexa demonstrates strong total write performance at 8 seconds. However, its total read latency of 13 seconds is higher when compared to some alternatives (e.g., MedRec at 2s, HealthRec-Chain at 4s). This characteristic stems primarily from Nexa's distinctive oracle design, where client-oracle communication, particularly through operations like *requestOracleAssistance()* and *submitReEncryptedShares()*, occurs via smart contract events. These interactions, crucial for maintaining immutable audit trails and protecting oracle identities without public exposure, are technically classified as write operations since emitting smart contract events involves logging payloads in transaction receipts stored on-chain. For that reason, these on-chain event emissions form an integral part of Nexa's read process and contribute to its overall read time.

The transition to the Avalanche mainnet is projected to significantly enhance Nexa's performance, particularly for its read operations. This projection is based on a key assumption: that

 doi:10.20944/preprints202602.0280.v1

the mainnet's documented sub-2-second finality will yield a proportional performance improvement over the observed Fuji testnet latencies. The mainnet's superior validator infrastructure and highly-optimized consensus mechanism are expected to deliver this faster finality. Since Nexa's read process incorporates multiple sequential on-chain smart contract events, the speed of these interactions directly correlates with network efficiency. Based on the mainnet's documented finality, these essential event emissions are projected to execute in under 2 seconds each, potentially reducing the total read operation time from 13 seconds to approximately 6 seconds (a reduction of about 54%). Similarly, the *addRecord()* operation, as a standard on-chain write transaction, is projected to see a similar benefit, decreasing from 8 seconds to around 5 seconds on the mainnet.

Competing solutions are also expected to benefit from mainnet environments. For instance, solutions built on platforms like Ethereum, which has transitioned to Proof-of-Stake (PoS), have seen improved transaction confirmation speeds. This primarily benefits their write operations, as these directly involve committing new data to the blockchain. As shown in Figure 6.3, mainnet projections suggest write latencies for EdgeHR, HealthRec-Chain, and MedRec could improve to approximately 16s, 18s, and 15s respectively. Their read latencies, which may rely on different mechanisms (such as direct state reads or off-chain data retrieval components that are less directly tied to new on-chain transaction finality for each read query), are expected to remain relatively stable or see less pronounced percentage gains compared to Nexa's read architecture, which relies on smart contract event emissions.

The projected mainnet performance, as illustrated in Figure 6.3 (bottom), positions Nexa as a particularly compelling solution, expected to balance read and write latencies at approximately 6 seconds and 5 seconds, respectively. This level of performance underscores its potential viability for healthcare applications where timely access to patient records is important. Furthermore, Nexa's adoption becomes even more attractive given Avalanche's custom subnet capabilities, which allow Nexa to isolate sensitive healthcare data processing from general network activity. This isolation ensures faster transactions by preventing congestion caused by unrelated network traffic, thereby maintaining consistency in latency and throughput. Additionally, the built-in sharding support offers a strong scaling mechanism, enabling Nexa to efficiently manage increasing volumes of patient data and access requests over time, maintaining high throughput even as transaction loads grow.

7. Discussion

7.1. Analysis of Results and Research Contributions

This thesis introduced Nexa, a blockchain architecture designed to address the critical need for secure and scalable EHR systems by balancing the inherent trade-offs defined by the blockchain trilemma. The core contribution lies in the integration of a high-performance public blockchain (Avalanche), decentralized storage (IPFS), and a distinct distributed threshold cryptography scheme (ChaCha20-Poly1305 with SSS and ECDH-based share distribution). The research aimed to answer how to optimize public blockchain characteristics for EHRs (RQ1) and how distributed key management could mitigate security vulnerabilities efficiently (RQ2).

The theoretical comparative analysis and further performance evaluation strongly suggest that Avalanche provides a suitable foundation for Nexa. Its high throughput (3,400+ TPS), rapid finality (~2 seconds), native sharding capabilities via subnets, and powerful EVM-compatible smart contract features (RBAC) directly address RQ1 by offering a compelling balance between scalability, security, and decentralization, remarkably outperforming many alternatives, especially legacy systems and earlier blockchain proposals reliant on less scalable platforms like Ethereum. While alternatives like NEAR showed higher theoretical TPS, Avalanche's mature ecosystem, EVM compatibility facilitating easier integration, and subnet architecture offered a more pragmatic and balanced choice for this initial architecture. The transaction costs on Avalanche, while higher than NEAR or Solana, remain orders of magnitude lower than Ethereum, presenting a cost-effective solution for healthcare operations.

 doi:10.20944/preprints202602.0280.v1

The distributed threshold cryptography mechanism implemented in Nexa directly addresses RQ2. By splitting the symmetric EHR encryption key (ChaCha20-Poly1305) into shares using Shamir's Secret Sharing (SSS) and distributing these shares encrypted via ECDH to multiple parties (patient, provider, decentralized oracles), Nexa effectively eliminates single points of failure inherent in centralized key management. Benchmarking confirmed the high efficiency of ChaCha20-Poly1305 and ECDH operations, adding minimal latency overhead (averaging ~20-25 ms for encryption/decryption cycles). This approach enhances security significantly, as compromising a single entity is insufficient to decrypt EHR data, leaving attackers needing to compromise multiple independent actors *($k \geq 3$)*. Furthermore, the access revocation mechanism, managed via smart contract logic without requiring re-encryption of historical data, proved to contribute to the scalability.

Performance benchmarks conducted on the Avalanche Fuji testnet provided encouraging results. Smart contract operations for RBAC and record management demonstrated acceptable latencies (typically 5-to-6-second average) and low gas costs, feasible for clinical workflows. Decentralized storage operations via IPFS/Filebase showed average latencies of ~1.3 seconds for read/write, suitable for EHR access, although outliers indicated potential latency increases with very large files, suggesting data segmentation strategies for production environments. The read latency involving oracle interaction was higher on the testnet (~13 s total) due to event emission and finality. However, projections for Avalanche mainnet suggest this could drop (to ~6 s), making Nexa even more competitive.

Despite the promising results, this study has limitations inherent to its scope. The evaluation relied on synthetic FHIR-compliant data and a testnet environment. Real-world clinical workflows, advanced user interface usability, and integration with existing hospital IT infrastructure were not assessed. The proof-of-concept focuses on the core architecture, leaving aspects like extensive legal compliance analysis and large-scale deployment details and challenges for future work.

7.2. Future Work

Building upon this foundation, several tracks for future work are identified:

- **Trusted Execution Environments (TEEs) [55]:** Exploring the integration of TEEs (e.g., Intel SGX, AMD SEV) for isolating the generation and handling of cryptographic keys and shares (both ephemeral ECDH keys and ChaCha20-Poly1305 key shares). This could further enhance security by protecting sensitive operations even if the host system is compromised.
- **Compromise-Triggered Key Rotation:** Developing a mechanism for rotating the ChaCha20-Poly1305 symmetric key for specific EHRs if a participating actor (e.g., a physician or oracle) is known to be compromised.
- **NEAR as an Alternative Platform:** Conducting a practical implementation and benchmarking study using the NEAR protocol. Given its high TPS and low fees identified in the comparative analysis, a direct comparison could reveal specific performance trade-offs and potentially offer an even more scalable or cost-effective alternative for certain deployment scenarios.
- **Post-Quantum Cryptography Integration [56]:** Investigating the integration of post-quantum cryptographic algorithms to ensure long-term security of the Nexa architecture against quantum computer threats, which represents an urgent priority for system sustainability.
- **Real-World Pilot Study:** Deploying Nexa in a controlled pilot program within a healthcare setting to evaluate its performance, usability, and integration capabilities under real-world conditions and gather feedback from clinicians and patients.
- **Interoperability with Legacy Systems:** Designing and evaluating standardized APIs and adapters (e.g., FHIR-based interfaces) to enable seamless integration of Nexa with existing hospital information systems and other healthcare IT infrastructure.

 doi:10.20944/preprints202602.0280.v1

8. Conclusion

This thesis addressed the persistent challenges of balancing security and scalability in EHR management by proposing Nexa, a distinct blockchain-based architecture. By strategically combining the high-performance Avalanche public blockchain, decentralized IPFS storage, and a distributed threshold cryptography scheme utilizing ChaCha20-Poly1305, Shamir's Secret Sharing, and Elliptic Curve Diffie-Hellman, Nexa aims to balance the blockchain trilemma, achieving scalability and decentralization without compromising security.

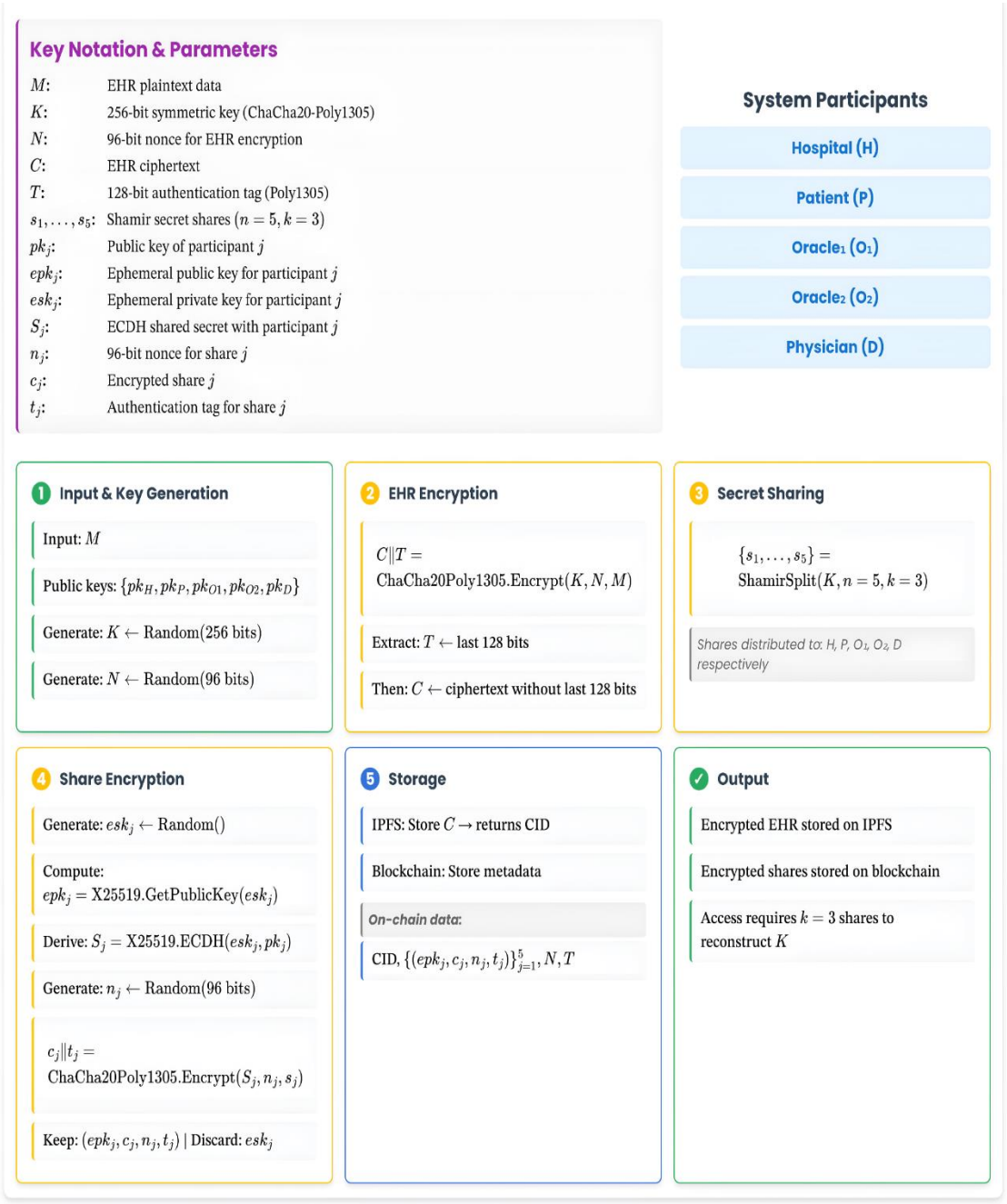
The research successfully identified optimal public blockchain characteristics for EHR systems (RQ1), favoring platforms like Avalanche with high TPS, low finality, sharding capabilities, and RBAC smart contract support. It also demonstrated how a distributed key management architecture that uses threshold cryptography and decentralized oracles, has the potential to effectively mitigate security vulnerabilities associated with single points of failure and enhance access control (RQ2). Performance evaluations using synthetic data on the Avalanche testnet confirmed the architecture's potential, showcasing efficient transaction processing, low operational costs, and excellent cryptographic performance that could be suitable for clinical interactions. Projected mainnet performance indicates Nexa could offer improvements over existing blockchain EHR proposals, particularly in write operation efficiency.

Even though Nexa has only been implemented as a proof-of-concept and evaluated under simulated conditions, it still revealed that thoughtful architectural design can harness the power of blockchain and advanced cryptography to create healthcare data management solutions that could be secure, scalable, patient-centric, and resilient against cyber threats. Future work focusing on TEE integration, key rotation strategies, alternative platforms like NEAR, post-quantum readiness, real-world pilots, and interoperability will be crucial in transitioning Nexa towards practical adoption. Eventually, this research contributes to the advancement of blockchain technology in healthcare, paving the way for more trustworthy and efficient EHR systems globally, particularly in an era of escalating healthcare cyber threats.

Statement of Originality and AI Usage: We, the authors, assert that this thesis is our original work. AI tools were used exclusively for grammar correction and language refinement. These tools did not contribute to the research, analysis, or conceptual development of the thesis.

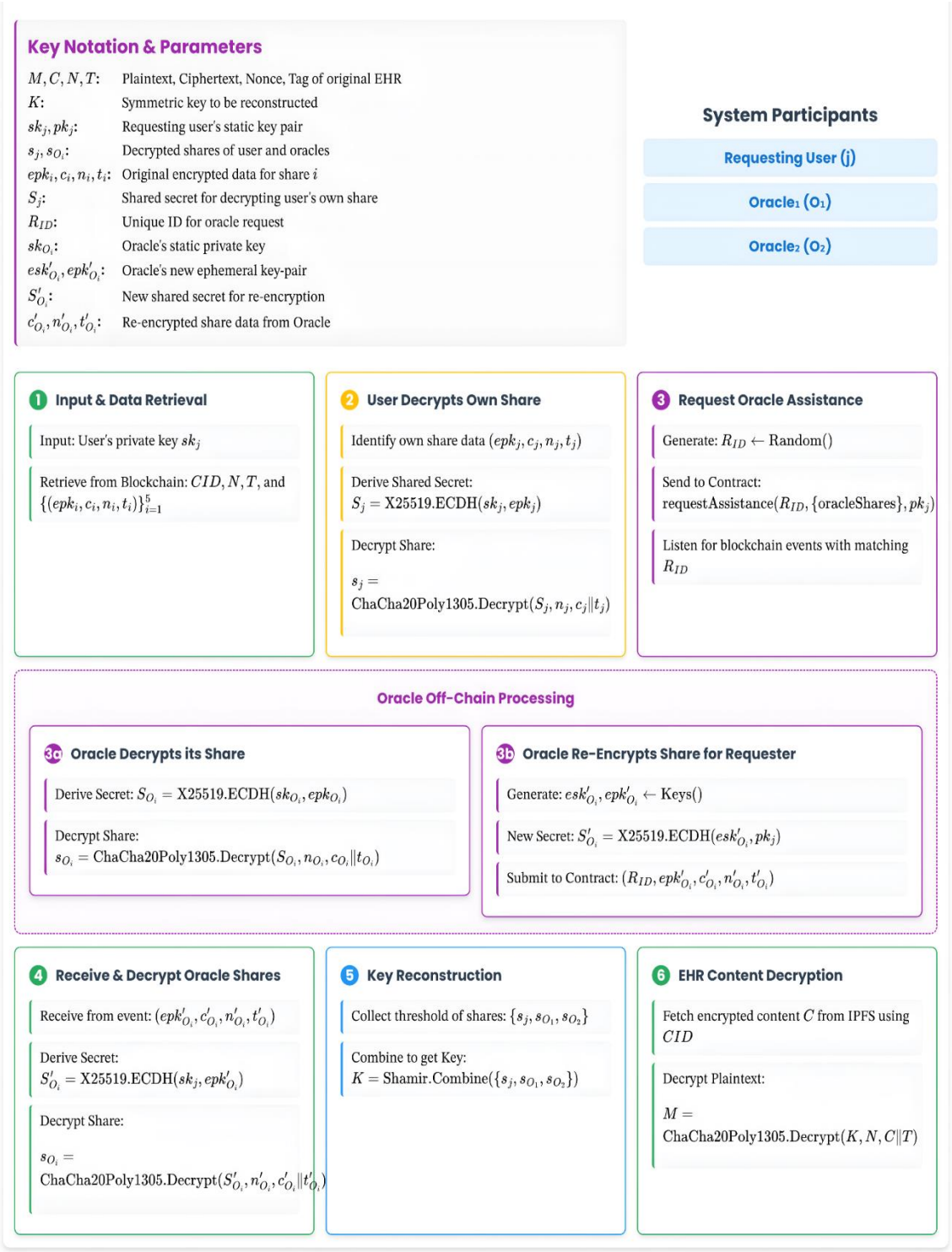
Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 February 2026 doi:10.20944/preprints202602.0280.v1

Appendix A — Nexa's Threshold Encryption Algorithm and Flow



 doi:10.20944/preprints202602.0280.v1

Appendix B — Nexa's Threshold Decryption Algorithm and Flow



References

1. D. Portela, D. Nogueira-Leite, R. Almeida, and R. Cruz-Correia, "Economic Impact of a Hospital Cyberattack in a National Health System: Descriptive Case Study," *JMIR Formative Research*, vol. 7, p. e41738, Jun. 2023, doi: 10.2196/41738.
2. T. Chen, X. Li, X. Luo, and X. Zhang, "Under-optimized smart contracts devour your money," *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Feb. 2017, doi: 10.1109/saner.2017.7884650.

 doi:10.20944/preprints202602.0280.v1

3. O. Konashevych, "Why 'Permissioned' and 'Private' are not Blockchains," *SSRN Electronic Journal*, 2019, doi: 10.2139/ssrn.3496468.
4. J. Walonoski *et al.*, "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *Journal of the American Medical Informatics Association*, vol. 25, no. 3, pp. 230–238, Aug. 2017, doi: 10.1093/jamia/ocx079.
5. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System." 2008. Accessed: May 11, 2025. [Online]. Available: https://bitcoin.org/bitcoin.pdf
6. F. Saleh, "Blockchain without Waste: Proof-of-Stake," *The Review of Financial Studies*, vol. 34, no. 3, pp. 1156–1190, Jul. 2020, doi: 10.1093/rfs/hhaa075.
7. S. H. Ammous, "Blockchain Technology: What is it Good for?," *SSRN Electronic Journal*, 2016, doi: 10.2139/ssrn.2832751.
8. Y. Han, Y. Zhang, and S. H. Vermund, "Blockchain Technology for Electronic Health Records," *International Journal of Environmental Research and Public Health*, vol. 19, no. 23, p. 15577, Nov. 2022, doi: 10.3390/ijerph192315577.
9. S. Mssassi and A. Abou El Kalam, "The Blockchain Trilemma: A Formal Proof of the Inherent Trade-Offs Among Decentralization, Security, and Scalability," *Applied Sciences*, vol. 15, no. 1, p. 19, Dec. 2024, doi: 10.3390/app15010019.
10. M. Baboi, "Security of Consensus Mechanisms in Blockchain," *Romanian Cyber Security Journal*, vol. 5, no. 2, pp. 45–53, Nov. 2023, doi: 10.54851/v5i2y202305.
11. V. Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform," ethereum.org. Accessed: May 11, 2025. [Online]. Available: https://ethereum.org/en/whitepaper/
12. M. Legault, "A Practitioner's View on Distributed Storage Systems: Overview, Challenges and Potential Solutions," *Technology Innovation Management Review*, pp. 32–41, Jul. 2021, doi: 10.22215/timreview/1448.
13. A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," *2016 2nd International Conference on Open and Big Data (OBD)*, Aug. 2016, doi: 10.1109/obd.2016.11.
14. D. Kumari, A. S. Parmar, H. S. Goyal, K. Mishra, and S. Panda, "HealthRec-Chain: Patient-centric blockchain enabled IPFS for privacy preserving scalable health data," *Computer Networks*, vol. 241, p. 110223, Mar. 2024, doi: 10.1016/j.comnet.2024.110223.
15. J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," arXiv.org. [Online]. Available: https://arxiv.org/abs/1407.3561
16. V. Mandarino, G. Pappalardo, and E. Tramontana, "A Blockchain-Based Electronic Health Record (EHR) System for Edge Computing Enhancing Security and Cost Efficiency," *Computers*, vol. 13, no. 6, p. 132, May 2024, doi: 10.3390/computers13060132.
17. A. Dubovitskaya *et al.*, "ACTION-EHR: Patient-Centric Blockchain-Based Electronic Health Record Data Management for Cancer Care," *Journal of Medical Internet Research*, vol. 22, no. 8, p. e13598, Aug. 2020, doi: 10.2196/13598.
18. E. Androulaki *et al.*, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," arXiv.org. [Online]. Available: https://arxiv.org/abs/1801.10228
19. B. Shen, J. Guo, and Y. Yang, "MedChain: Efficient Healthcare Data Sharing via Blockchain," *Applied Sciences*, vol. 9, no. 6, p. 1207, Mar. 2019, doi: 10.3390/app9061207.
20. L. D. Costa, B. Pinheiro, W. Cordeiro, R. Araújo, and A. Abelém, "Sec-Health: A Blockchain-Based Protocol for Securing Health Records," *IEEE Access*, vol. 11, pp. 16605–16620, 2023, doi: 10.1109/access.2023.3245046.
21. E. A. Abdullah, "A Hybrid Algorithm for Encrypting Electronic Health Record Using Blockchain in a Cloud Computing Environment," *International Journal of Intelligent Systems and Applications in Engineering*, no. 22s. pp. 903–912, Jul. 09, 2024. Accessed: May 11, 2025. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/6573
22. K. Sekniqi, D. Laine, Stephen Buttolph, and E. G. Sirer, "Avalanche Platform." AvaLabs, Jun. 30, 2020. Accessed: May 11, 2025. [Online]. Available: https://www.avalabs.org/whitepapers
23. A. Abbasi, "Synthetic EHRs for Benchmarking System Performance," Kaggle. Accessed: May 11, 2025. [Online]. Available: https://www.kaggle.com/dsv/11614066

 doi:10.20944/preprints202602.0280.v1

24. C. Suter-Crazzolara, "Better Patient Outcomes Through Mining of Biomedical Big Data," *Frontiers in ICT*, vol. 5, Dec. 2018, doi: 10.3389/fict.2018.00030.
25. B. Hou and F. Chen, "A Study on Nine Years of Bitcoin Transactions: Understanding Real-world Behaviors of Bitcoin Miners and Users," *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1031–1043, Nov. 2020, doi: 10.1109/icdcs47774.2020.00091.
26. A. Jain, C. Jain, and K. Krystyniak, "Blockchain transaction fee and Ethereum Merge," *Finance Research Letters*, vol. 58, p. 104507, Dec. 2023, doi: 10.1016/j.frl.2023.104507.
27. B. L. Y. Quan *et al.*, "Recent Advances in Sharding Techniques for Scalable Blockchain Networks: A Review," *IEEE Access*, vol. 13, pp. 21335–21366, 2025, doi: 10.1109/access.2024.3523256.
28. K. Croman *et al.*, "On Scaling Decentralized Blockchains," in *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 106–125. [Online]. Available: https://doi.org/10.1007/978-3-662-53357-4_8
29. G. Kaur and C. Gandhi, "Scalability in Blockchain: Challenges and Solutions," in *Handbook of Research on Blockchain Technology*, Elsevier, 2020, pp. 373–406. [Online]. Available: https://doi.org/10.1016/b978-0-12-819816-2.00015-0
30. J. Liu, W. Zheng, D. Lu, J. Wu, and Z. Zheng, "Understanding the Decentralization of DPoS: Perspectives From Data-Driven Analysis on EOSIO," arXiv.org. [Online]. Available: https://arxiv.org/abs/2201.06187
31. S. Joshi, "Feasibility of Proof of Authority as a Consensus Protocol Model," arXiv.org. [Online]. Available: https://arxiv.org/abs/2109.02480
32. "Solidity 0.8.30 documentation," Solidity. Accessed: May 11, 2025. [Online]. Available: https://docs.soliditylang.org/en/stable/
33. "Filebase Documentation," Filebase. Accessed: May 11, 2025. [Online]. Available: http://docs.filebase.com/
34. D. Vorick and L. Champine, "Sia: Simple Decentralized Storage." Nebulous Inc., Nov. 29, 2014. Accessed: May 11, 2025. [Online]. Available: https://sia.tech/sia.pdf
35. "Filecoin: A Decentralized Storage Network." Protocol Labs, Jul. 19, 2017. Accessed: May 11, 2025. [Online]. Available: https://filecoin.io/filecoin.pdf
36. S. Williams, A. Kedia, L. Berman, and S. Campos-Groth, "Arweave: The Permanent Information Storage Protocol." Arweave, Dec. 26, 2023. Accessed: May 11, 2025. [Online]. Available: https://www.arweave.org/files/arweave-lightpaper.pdf
37. A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979, doi: 10.1145/359168.359176.
38. Y. Nir and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols," RFC Editor, Jun. 2018. Accessed: May 11, 2025. [Online]. Available: https://doi.org/10.17487/rfc8439
39. R. K. Muhammed *et al.*, "Comparative Analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for Image Encryption," *Kurdistan Journal of Applied Research*, vol. 9, no. 1, pp. 52–65, May 2024, doi: 10.24017/science.2024.1.5.
40. D. J. Bernstein, "Curve25519: New Diffie-Hellman Speed Records," in *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 207–228. [Online]. Available: https://doi.org/10.1007/11745853_14
41. F. Mallouli, A. Hellal, N. Sharief Saeed, and F. Abdulraheem Alzahrani, "A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms," *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pp. 173–176, Jun. 2019, doi: 10.1109/cscloud/edgecom.2019.00022.
42. "TypeScript Documentation," TypeScript. Accessed: May 11, 2025. [Online]. Available: https://www.typescriptlang.org/docs/
43. "Hardhat Documentation," Hardhat, Nomic Foundation. Accessed: May 11, 2025. [Online]. Available: https://hardhat.org/docs
44. A. Abbasi and M. N. Humeidi, "NexaEHR Prototype," GitHub. Accessed: May 11, 2025. [Online]. Available: https://github.com/alcompilor/nexa

 doi:10.20944/preprints202602.0280.v1

45. "MetaMask Developer Documentation," MetaMask. Accessed: May 11, 2025. [Online]. Available: https://docs.metamask.io/
46. "The NEAR White Paper." Accessed: May 11, 2025. [Online]. Available: https://pages.near.org/papers/the-official-near-white-paper
47. A. Yakovenko, "Solana: A new architecture for a high performance blockchain v0.8.13." Accessed: May 11, 2025. [Online]. Available: https://solana.com/solana-whitepaper.pdf
48. G. Wood, "Polkadot: Vision for a Heterogeneous Multi-chain Framework." Accessed: May 11, 2025. [Online]. Available: https://polkadot.com/papers/Polkadot-whitepaper.pdf
49. "Cardano Docs," Cardano. Accessed: May 11, 2025. [Online]. Available: https://docs.cardano.org/
50. "BNB Smart Chain Whitepaper," GitHub. Accessed: May 11, 2025. [Online]. Available: https://github.com/bnb-chain/whitepaper/blob/master/WHITEPAPER.md
51. TRON DAO, "Tron: Advanced Decentralized Blockchain Platform," no. 2.0. Dec. 10, 2018. Accessed: May 11, 2025. [Online]. Available: https://tron.network/static/doc/white_paper_v_2_0.pdf
52. H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts." 27th USENIX Security Symposium, Aug. 2018. Accessed: May 11, 2025. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-kalodner.pdf
53. "Optimism Docs," Optimism. Accessed: May 11, 2025. [Online]. Available: https://docs.optimism.io/
54. A. Abbasi, "NexaEHR Benchmarks," Kaggle. Accessed: May 15, 2025. [Online]. Available: https://www.kaggle.com/dsv/11826992
55. X. Li, B. Zhao, G. Yang, T. Xiang, J. Weng, and R. H. Deng, "A Survey of Secure Computation Using Trusted Execution Environments," arXiv.org. [Online]. Available: https://arxiv.org/abs/2302.12150
56. D. J. Bernstein, "Introduction to post-quantum cryptography," in *Post-Quantum Cryptography*, D. J. Bernstein, J. Buchmann, and E. D. Goodman, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–14. Accessed: May 11, 2025. [Online]. Available: https://doi.org/10.1007/978-3-540-88702-7_1