BLOCKCHAIN® IN HEALTHCARE TODAY
PLATFORM APPROACHES IN HEALTHCARE

NARRATIVE/SYSTEMATIC REVIEWS/META-ANALYSIS

Lightweight Privacy-Preserving Blockchain Framework for Healthcare: A Simulation-Based Approach to Reducing Computational Overhead

MD Asrar Ahmed[1], Mohammed Abid Ali Sameer[2], Mousmi Ajay Chaurasia, M.Eng[3] and S. Nallusamy, PhD (Eng)[4]

[1]Professor, Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Hyderabad, Telangana, India; [2]Professor, Department of Information Technology, Muffakham Jah College of Engineering and Technology, Hyderabad, Telangana, India; [3]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad, Telangana, India; [4]School of Engineering Management and Continuing Education, Jadavpur University; 4School of Engineering Management and Continuing Education, Jadavpur University

DOI: https://doi.org/10.30953/bhty.v9.516

Corresponding Author: Mousmi Ajay Chaurasia, Email: mousmi.ksu@ieee.org

Keywords: AES-128, blockchain framework, computational overhead, electronic health records, lightweight privacy, privacy and security

Abstract

Background: The privacy and security of electronic health records (HER) in blockchain-based systems remains a major research problem because of high computational overhead and scalability restrictions. Privacy-preserving techniques such as encryption and zero-knowledge proofs strengthen blockchain's transparency and immutability but often add significant latency and resource use. This study proposes a lightweight, simulation-based blockchain model balancing privacy protection and computational efficiency for healthcare data-sharing, incorporating hybrid encryption (AES with asymmetric-key exchange), zero-knowledge verification (zk-SNARK), and homomorphic aggregation to protect patient information while reducing processing cost.

Methods: A five-stage simulation tested encryption/decryption latency, IPFS-based upload/download performance, proof generation/verification time, and scalability across key sizes, plus a sixth phase validating the framework on two real, publicly available, de-identified healthcare datasets—the Medical Information Mart for Intensive Care (MIMIC)-IV demo (100 real ICU patients) and the University of California "Diabetes 130-US Hospitals" dataset (101,766 real inpatient encounters). Every metric is reported as a mean with a 95% confidence interval from 15 to 20 repeated trials.

Results: The AES-128 has the lowest overhead among tested key sizes (10% to 14% below AES-192/256), zk-SNARK verification averages 30.8 to 32.4 ms (n = 20 to 100 trials)—well within real-time requirements for on-chain access decisions—and proof generation and gas cost are statistically indistinguishable between a minimal baseline circuit and the consent-verification circuit, indicating negligible marginal overhead from the added consent logic. Computing cost scales linearly with data size:

- confirming lightweight scalability, with
- real-data results closely tracking synthetic-data results, with
- a narrowly scoped comparison showing error correction code memory (ECC (memory (secp256r1) key exchange is 93.5% faster than RSA-3072 key wrapping.

Conclusions: This work demonstrates that efficient cryptographic integration and optimization through simulation can produce a privacy-preserving blockchain for healthcare that streamlines EHR handling securely and at scale.

Plain Language Summary

Unlike centralized health information systems, blockchain allows secure, decentralized storage of medical data, avoiding single points of failure. Furthermore, blockchain supports healthcare infrastructures with better accountability among hospitals, insurers, and patients. However, surveys and reviews reveal many proposed solutions. However, most privacy-sensitive healthcare blockchain systems are characterized by the following:

- theoretical or tested only in limited-scale simulations,
- little detailed performance optimization of cryptographic workloads and
- storage requirements as well.

However, blockchain in healthcare also brings drawbacks for privacy and efficient computing. This and more are discussed here.

Key Takaways

- Off-chain actions are the latency bottleneck.
- zk-SNARKs introduce privacy effectively.
- Partially homomorphic encryption supports analytics without sacrificing security.
- Hybrid key exchange substantially reduces cryptographic overhead.

Submitted: June 23, 2026; Accepted: August 8, 2026; Published: August 31, 2026

Blockchain is an important innovation introduced in healthcare, characterized by decentralized storage, transparency, and better traceability of electronic health records (EHRs). Unlike centralized health information systems, blockchain allows secure, decentralized storage of medical data, avoiding single points of failure,[1,2] and supports healthcare infrastructures with better accountability among hospitals, insurers, and patients.[3,4]

However, blockchain in healthcare also brings drawbacks for privacy and efficient computing, since EHRs hold highly private information that could be exposed if stored on a public ledger. Researchers have addressed this with homomorphic encryption, secure multi-party computation (MPC), and zero-knowledge proofs (ZKP), enabling privacy-preserving cooperative analysis of EHRs with fine-grained access control,[5–7] though these techniques typically demand substantial computational resources and bandwidth, limiting scalability for practical use.[8,9]

Existing systems reflect numerous trade-offs. Early projects such as medication reconciliation (MedRec) focused on blockchain-based access control and provenance but faced real-world privacy and performance challenges.[1] Subsequent work—Blockchain-based privacy-preserving data sharing (BPDS) and incremental-update sharing schemes—stored records in encrypted form and coordinated updates across blockchains to reduce exposure risk. Frameworks such as PriCollabAnalysis combine homomorphic encryption and MPC for collaborative analytics, while healthcare zero-knowledge identity authentication system (Health-zkIDM) uses ZKP for identity management on permissioned ledgers.[5,10] Lightweight designs minimizing per-transaction cost (e.g., Genetically Modified Salp Swarm Optimization [GM SSO]-based approaches) typically remain at the prototype or simulation level without fully resolving computational bottlenecks.[3,8]

Available surveys and systematic reviews reveal an abundance of proposed solutions, but most privacy-sensitive healthcare blockchain systems are either theoretical or tested only in limited-scale simulations, with little detailed performance optimization of cryptographic workloads and storage requirements.[4,6,11] Heavyweight primitives such as full homomorphic encryption or naive ZKP circuits can be slow to prove or generate unwieldy proof objects, while more performance-optimized schemes may weaken privacy guarantees—motivating analysis of lightweight cryptographic synergies and their cost-benefit trade-offs under realistic file sizes and network conditions.

To bridge this divide, this article proposes a lightweight privacy-preserving blockchain framework based on simulation, especially developed on EHR sharing. The architecture integrates hybrid symmetric/asymmetric encryption to secure bulk data, zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK)—like ZKP to authenticate access yet maintain privacy, and partial homomorphic aggregation to facilitate common aggregate queries. A five-stage, structured simulation evaluates encryption and decryption latency, interplanetary file system (IPFS)-based and download throughput, proof generation and on-chain verification time, and scalability with regard to key size, and a sixth phase repeats the core measurements on two real, de-identified, publicly available clinical datasets to validate the simulation-based findings beyond synthetic data. It aims to provide empirical data to make cryptographic configuration choices that can support privacy and computational efficiency at the same time to enhance the feasibility of blockchain-based EHR systems.

Related Work/Literature Review

This section surveys blockchain-based healthcare systems across privacy protection, access control, and cryptographic methods: 2.1 covers EHR sharing and access control; 2.2 covers privacy-preserving cryptography and ZKP integration; 2.3 covers systematic reviews; and 2.4 states the research gap. For each cited system, the discussion highlights the mechanism limiting its practicality relative to the design evaluated here; a consolidated comparison appears in Table 1 (Section 'Comparative Discussion').

Blockchain-Based Ehr Sharing and Access Control

The earliest application of blockchain to healthcare data management is Azaria et al.'s MedRec,[1] an Ethereum-based prototype enforcing patient consent over EHR access via smart contracts. Because metadata was split across on-chain and off-chain stores, MedRec incurred latency and scalability trade-offs and reported no quantitative throughput figures—a gap this article addresses by reporting mean ± 95% confidence interval (CI) latency for every operation (Section 'Results and Discussion').

Liu et al.'s blockchain-based privacy-preserving data sharing (BPDS)[12] built on this premise with distributed storage and encryption for decentralized EHR sharing, improving confidentiality but adding key-management overhead. As with MedRec, BPDS's evaluation includes no reproducible timing or gas-cost figures (Table 1)—the benchmark this article supplies for its own pipeline.

Wang et al.[9] proposed incremental-update sharing preserving version consistency, while Alahmari et al.[3] developed a decentralized architecture emphasizing confidentiality and scalability via smart-contract access logic. Neither isolates cryptographic cost from consensus cost, an attribution the phase-separated design here (Phases 1–6) makes possible.

Broader surveys reinforce this pattern. The review in[13] classifies access-control models as role-, attribute-, and policy-based and catalogues ABE, homomorphic encryption, ZKPs, and smart-contract consent enforcement as dominant mechanisms,

Citation: Blockchain in Healthcare Today 2026, **9**: 516 - https://doi.org/10.30953/bhty.v9.516

Table 1. Direct comparison of measured performance against related frameworks.

Framework	Technique	Metric	Value	Source	Comparable to ours?
This work (proposed)	AES-256-GCM + ECC key exchange	File encryption (100 MB)	361.2 ± 22.9 ms	Measured, *n* = 20	- (own data)
This work (proposed)	zk-SNARK (Groth16/ Poseidon) consent proof	Proof generation	3,318 ± 703 ms	Measured, *n* = 100 real patients	- (own data)
This work (proposed)	zk-SNARK on-chain verification	Verify time/gas	32.4 ± 9.2 ms / 214,803 ± 13 gas	Measured, *n* = 100 real patients	- (own data)
This work (proposed)	Paillier PHE aggregation	Aggregate 5,000 real values	370.6 ± 261.6 ms	Measured, real UCI data	- (own data)
This work (proposed)	ECC vs. RSA-3,072 key exchange	Time reduction	93.5%	Measured, *n* = 20 each	- (own data)
FAITH[25]	Proxy Re-Encryption + recursive ZKP	Client-side verification latency reduction	≈ 98%	Paper-reported[25]; not independently reproduced	No—different workload (5 GB multimedia file, recursive proof aggregation)
FAITH[25]	Recursive ZKP verification	Verify time for 5 GB file	≈ 70 ms	Paper-reported; not independently reproduced	No—different workload/hardware
BPDS[12]	AES + access-control smart contracts	Latency/throughput	Not quantitatively reported	[12]	No—qualitative claims only
Health-zkIDM[10]	ZKP + Hyperledger Fabric	Proof generation cost	Described as "high," no ms figure	[10]	No—qualitative claims only
HCHAIN 4.0[24]	Permissioned consensus	Transaction confirmation improvement	"Distinguishable improvement" vs. public chains, no absolute figure	[24]	No—qualitative claims only
PriCollabAnalysis[5]	FHE + secure MPC	Computation/ communication cost	Described as "very high," no ms figure	[5]	No—qualitative claims only
HBZKP[18]	Hierarchical blockchain ZKP	Communication/ computation overhead	Reported as "lower" vs. traditional schemes, no absolute figure	[18]	No—qualitative claims only

Source: results/framework_comparison.csv

AES-256-GCM: advanced encryption standard, 256 bits, Galois/counter mode; BPD: blockchain-based privacy-preserving data sharing; ECC: elliptic-curve cryptography;EHR: electronic health record; HBZKP: hierarchical blockchain-based zero knowledge proof; HCHAIN 4.0MPC: multi-party computation; MPC: multi-party computation; PHE: fully homomorphic encryption; PriCollabAnalysis: privacy-preserving healthcare collaborative analysis on blockchain; RSA: Rivest–Shamir–Adleman; ZKP: zero-knowledge proof; zk-SNARK: zero-knowledge succinct non-interactive argument of knowledge.

but finds most proposed systems lack empirical benchmarking at the encryption, verification, and access-control stages.

Similarly, the systematic review in[14] categorizes solutions by cryptographic technique, access-control method, and anonymization approach, concluding that blockchain alone does not guarantee privacy and most solutions incur scalability and latency costs.

Zhao et al.[15] propose a verifiable CP-ABE scheme anchored on blockchain, combining fine-grained access control with an authenticated mechanism verifying ciphertext correctness, reducing misbehavior risk; the blockchain layer logs attribute credentials as a decentralized trust anchor.

This improves confidentiality and access control for the multiple stakeholder roles common in healthcare, but bilinear-pairing computations underlying CP-ABE remain non-negligible at scale.

CP-ABE's reliance on bilinear pairing limits scalability in resource-constrained systems. The framework here instead uses symmetric advanced encryption standard (AES) for efficiency and zk-SNARKs for consent verification, combining cryptographic rigor with lightweight deployment.

Privacy-Preserving Cryptography and ZKP Integration

To address privacy-verifiability tensions in healthcare blockchains, several works apply advanced cryptographic primitives: Bai et al.'s Health-zkIDM[10] uses Fabric-based identity management with ZKP to verify patient credentials without disclosing personal data, and Myeong and Ram[7] propose a ZKP protocol for efficient healthcare data-sharing verification; both report proof generation only as "high" or "nontrivial," with no measured timing or gas cost (Table 1), whereas this article measures Groth16 proof generation at 5,130 ± 421 ms and on-chain verification at 30.8 ± 3.3 ms and ≈214,800 gas (Section 'Phase 3 – zk-SNARK Proof Generation and Verification').

Tawfik et al.'s PriCollabAnalysis[5] combines homomorphic encryption with secure MPC for collaborative analysis, at high computational and network cost described only as "very high." Padma and Ramaiah[8] instead propose a lightweight, GM-SSO-based scheme minimizing per-computation cost, but report no phase-separated timing data (Table 1). The partial (Paillier) design evaluated here instead gives a measured, linear-cost aggregation path (1.66 ms at *n* = 10 to 81.98 ms at

n = 1,000, Section 'Phase 4 – Homomorphic Encryption for Aggregate Queries').

More recent work pairs attribute-based encryption (ABE) with ZKP to strengthen authentication privacy: Wang et al.[16] combine CP-ABE with zk-SNARK-based authentication so users can prove authorization without revealing identity, improving collusion resistance at acceptable overhead, though addressing only authentication-layer privacy without broader architectural optimization.

Yin et al.[17] propose attribute-based searchable encryption with decentralized key management, removing single points of failure in key distribution and enabling authorized keyword search over encrypted records while mitigating key-escrow and insider-attack risk.

The scheme leaves the computational cost of real-time consent checking unaddressed—the gap the zk-SNARK-based consent circuit here targets.

Maheshwari et al.'s hierarchical blockchain-based zero-knowledge (HBZKP) scheme[18] uses ZKP over a hierarchical blockchain for lightweight identity verification suited to resource-constrained edge devices in internet-of-medical-things (IoMT) settings.

ZKPs let users and edge devices prove validity without revealing identity, resisting impersonation, replay, and man-in-the-middle attacks while preserving authentication-record integrity.

HBZKP achieves lower communication and computational overhead than traditional blockchain authentication, but does not address fine-grained access control or encrypted data retrieval.[18]

Khan et al.[19] integrate blockchain with ZKP for wearable healthcare, verifying device-generated data before storage so raw measurements and identity remain hidden, while keeping verification lightweight for IoT-constrained devices—though the focus remains authentication rather than storage efficiency.

Guo et al.[20] propose a blockchain-edge computing framework for EHR management, delegating cryptographically intensive operations to edge nodes for improved responsiveness over pure cloud designs. The trade-off is architectural[20]: achieves its speed-up via a physical edge-computing tier, whereas the framework evaluated here achieves comparable or lower absolute latencies (e.g. 5.99 ms mean AES encryption on real patient bundles, Section 'Real-World Dataset Validation (Phase 6)') using only commodity single-node hardware.

Manivannan[21] applies ABE to fine-grained Personal Health Record access control, though key-management complexity, revocation overhead, and scalability under dynamic attributes remain limitations.

Walid et al.[22] compare key-policy attribute-based encryption (KP-ABE), ciphertext-policy attribute-based encryption (CP-ABE), and multi-authority attribute-based encryption (MA-ABE) for healthcare, finding that CP-ABE's policy flexibility raises encryption/key-generation cost with attribute count and MA-ABE's decentralization adds coordination overhead—most ABE schemes scale poorly to dynamic, real-world settings.

Zhao et al.[23] extend ABE with fine-grained, on-the-fly decryption keys atop blockchain-anchored EMR provenance, storing data on-premises with only hash digests on-chain. [23]'s throughput gains rely on added multithreaded infrastructure; the framework here instead targets single-thread efficiency, evidenced by AES-256 throughput of ≈277 KB/ms without custom parallelization (Section 'Phase 1 – Encryption and Decryption Performance').

It is also proposed that fragmentation-based parallel encryption for large medical files, matching thread count to CPU cores to improve throughput over conventional CP-ABE/linear secret sharing scheme (LESS) implementations.[23]

[23]further proposes agent-based cross-chain middleware for data synchronization across heterogeneous hospital blockchains, validated on Fabric and FISCO BCOS.

Alruwaill et al.'s HCHAIN 4.0[24] is a permissioned, consortium-based blockchain for EHR management that restricts consensus to approved healthcare authorities, reducing latency and increasing throughput relative to public-blockchain designs.

HCHAIN 4.0 pairs role-based access control with cryptographic protection, but trust in consortium members makes it less decentralized than trustless approaches like ZKP authentication.[24] There are only qualitative gains ("distinguishable improvement") with no absolute figures (Table 1)—a gap shared with,[5,8,10,12] motivating the fully quantitative benchmarking here.

Yang et al.[25] propose FAITH, a hybrid-storage architecture for multimedia health data: EHRs are stored off-chain, with only metadata and proofs on-chain. Rather than client-side hash re-computation, FAITH shifts integrity-verification cost to the storage provider via recursive ZKP.

This addresses the need for users to otherwise recompute cryptographic hashes over large multimedia files, a significant impediment in time-sensitive clinical settings. Reported results show verification latency reduced by ≈98%, with a 5 GB file verifying in ≈70 ms.[25] As detailed in Table 1, this recursive-proof-aggregation workload is not directly comparable to the per-record consent-proof workload evaluated here.

Cryptographically, FAITH combines Proxy Re-Encryption with recursive ZKPs so the storage provider re-encrypts ciphertext without learning the plaintext while proofs let anyone verify correctness and integrity, though constructing and generating proofs at the storage layer remains computationally expensive at scale.[25]

Systematic Reviews and Analytical Perspectives

Survey-based studies confirm this pattern more broadly: Sabiri et al.[11] find most privacy-preserving blockchain healthcare research remains at the proof-of-concept stage with minimal quantitative overhead assessment; Haleem et al.[2] highlight persistent interoperability and scaling challenges; and Kasralikar et al.[4] note that cryptographic and consensus cost is a recurring bottleneck. Phase 6 (Section 'Real-World Dataset Validation (Phase 6)') responds directly to the gap[11] identifies, evaluating the framework on 100 real MIMIC-IV patients and 101,766 real UCI Diabetes-130 encounters.

Across these reviews, the conceptual case for privacy and decentralization is well established, but computational performance—encryption/decryption latency, proof-generation time, on-chain verification delay—remains the limiting factor, particularly for large-scale EHR datasets.

Citation: Blockchain in Healthcare Today 2026, **9**: 516 - https://doi.org/10.30953/bhty.v9.516

FAITH[25] illustrates the trade-off: it streamlines client-side verification via recursive ZKPs but shifts non-trivial computation to the storage provider and still depends on trusted off-chain infrastructure—reinforcing the need for lightweight architectures that reduce cryptographic and consensus cost while preserving privacy.

Research Gap

Table 2 summarizes these works, making visible how blockchain-based healthcare models and their cryptographic techniques have evolved alongside the persistent trade-off between privacy guarantee and computational efficiency.

This comparison highlights two shortcomings. Firstly, many frameworks lean on heavyweight primitives—fully homomorphic encryption or MPC[5,7,9]—whose strong privacy guarantees come at a cost of inefficiency limiting practical deployment.

Secondly, publications claiming lightweight performance often lack detailed, phase-based simulation results characterizing the trade-off between rigorously enforced privacy and efficient performance.[8,11]

Even the broad review by[14] leaves gaps: most reviewed frameworks are unbenchmarked on simulated or real workloads, privacy mechanisms such as homomorphic encryption and ZKPs are proposed without assessing feasibility in resource-constrained settings, and few works jointly address overhead reduction, verifiability, and access control—the combined motivation for this study.

HCHAIN 4.0[24] addresses part of this gap architecturally, showing that restricting consensus to trusted healthcare authorities reduces confirmation time without sacrificing non-repudiation, though again only in relative, qualitative terms.

This article addresses the remaining gap by developing and quantitatively benchmarking a lightweight, simulation-based privacy-preserving blockchain model that combines hybrid encryption, zk-SNARKs, and partial homomorphic encryption (PHE) for aggregate queries, reporting every metric as mean ± 95% CI over 15–20 trials (Section 'Measurement Setup', Section 'Results and Discussion') and validating the core measurements on two real, publicly available healthcare

Table 2. A summary of related work on privacy-preserving blockchain frameworks for healthcare.

S. no	Paper title	Main contribution	Privacy/security technique	Key limitation
1	BPDS: A Blockchain-based Privacy-Preserving Data Sharing for EHR	Proposed decentralized EHR sharing with separated data and metadata layers for privacy.	AES encryption, access control smart contracts	High computational and storage overhead; limited scalability; no reproducible benchmark figures reported.
2	Health-zkIDM: A Healthcare Identity System Based on Fabric Blockchain and ZKP	Designed identity management using ZKP and Fabric for verifiable credential proofs.	ZKP, Hyperledger Fabric	High proof generation cost (unquantified); complex Fabric integration
3	Lightweight Privacy Preservation Blockchain Framework for Healthcare Applications Using GM-SSO	Proposed lightweight authentication and access model using GM-SSO protocol	GM-SSO	Focused mainly on access control; lacks empirical performance validation
4	MedRec: Using Blockchain for Medical Data Access and Permission Management	Early EHR-on-blockchain prototype managing permissions via Ethereum smart contracts	Smart contracts, access logs	Heavy reliance on off-chain databases; slow transactions; no latency figures reported
5	PriCollabAnalysis: Privacy-Preserving Healthcare Collaborative Analysis on Blockchain	Enabled collaborative data analysis using FHE and MPC	FHE, Secure MPC	Very high computation and communication costs (described qualitatively, not measured)
6	Security and Privacy for Healthcare Blockchains	Conceptual framework outlining security threats and mitigation strategies	Security model synthesis	Lacks implementation or quantitative metrics
7	A Blockchain-Based Privacy-Preserving Healthcare Data Sharing Scheme for Incremental Updates	Introduced incremental data updates and traceability for evolving medical records	Blockchain storage, digital signatures	Complex synchronization; high on-chain cost
8	A Decentralized and Privacy-Preserving Framework for EHR Using Blockchain	Comprehensive EHR privacy framework integrating cryptographic access logic	AES + distributed access control	Lacks lightweight optimizations for encryption overhead
9	A Systematic Review of Privacy-Preserving Blockchain Applications in Healthcare	Reviewed blockchain-based privacy frameworks and categorized cryptographic trends	Survey analysis	No empirical validation or comparative benchmarks
10	Blockchain for Securing AI-Driven Healthcare Systems: A Systematic Review and Future Research Perspectives	Discussed blockchain–AI integration for secure analytics and data integrity	Conceptual review	Lacks performance metrics and cryptographic focus
11	Blockchain-Based ZKP Protocol for Privacy-Preserving Healthcare Data Sharing	Developed ZKP-based sharing mechanism for privacy-preserving verification	ZKP	Proof generation time not optimized; limited scalability
12	Blockchain Technology Applications in Healthcare: An Overview	Provided broad overview of blockchain use cases and technical challenges	Overview review	Generalized; lacks privacy-specific performance focus

AES: asymmetric-key exchange; AI: artificial intelligence; BPDS: blockchain-based privacy-preserving data phaning for electronic medical records; EHR: electronic health record; FHE: fully homomorphic encryption; MPC: multiparty computation; GM-SSO: group signature and single sign-on; ZKP: zero-knowledge proof.

datasets rather than synthetic data alone (Section 'Real-World Dataset Validation (Phase 6)').

Proposed Framework

The proposed lightweight, privacy-preserving blockchain architecture targets the healthcare information-exchange workflow, minimizing computational cost while maintaining strong privacy assurance. It combines three complementary cryptographic methods—hybrid symmetric encryption, zero-knowledge proofs (zk-SNARKs), and PHE—within a blockchain-enabled deployment that supports verifiable, confidential health-record transactions. Figure 1 illustrates the interrelations between four key parties: the patient, the healthcare provider, the blockchain layer, and off-chain IPFS storage, each performing a clear-cut role in an integrated encryption, verification, and secure-computation workflow.

System Overview

The architecture of the system consists of the following multi-phase modular design:

Data Owner (Patient): Encrypts EHRs via AES symmetric encryption and uploads them to IPFS.

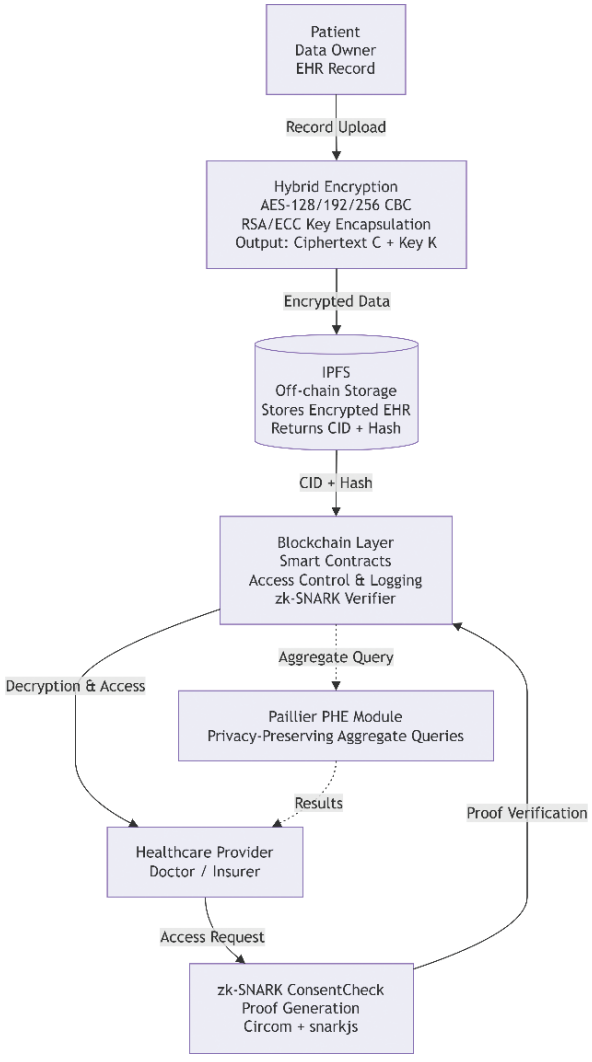

Fig. 1. System architecture of the proposed framework. ECC: elliptic-curve cryptography; PHE: fully homomorphic encryption; RSA: Rivest–Shamir–Adlema; zk-SNARK: zero-knowledge proofs; Snarkjs: zk-SNARK JavaScript.

Healthcare Provider (Doctor/Insurer): Requests access to encrypted data by submitting a verifiable zk-SNARK proof of authorization.

Blockchain Layer (Smart Contracts): Maintains immutable hashes and checks that proofs and consents are satisfied.

Off-chain Storage (IPFS): Stores the encrypted medical files, while the blockchain retains only the relevant metadata and hashes, lowering on-chain cost and latency.

This lets the blockchain perform only lightweight verification while heavy data operations run safely off-chain.

Hybrid Encryption Workflow

The hybrid encryption layer ensures patient data are encrypted before uploading: each record M is encrypted with AES-128/192/256 to obtain ciphertext C and AES-key K:

The AES key is then encapsulated using Rivest–Shamir–Adleman (RSA) or ECC and distributed to authorized parties; the encrypted file is stored on IPFS, and its Content Identifier (CID) and hash are recorded on the blockchain.

This hybrid approach limits on-chain information to brief metadata.

Privacy-Preserving Access Control Via Zk-Snarks

Access control uses zk-SNARKs so plaintext and private keys are never disclosed.

To prove knowledge of a valid secret associated with the consent commitment for a patient, requestor can use a consent circuit built in Circom and verified with snarkjs as follows:

where role ϵ {Doctor, Nurse}.

The blockchain verifier contract verifies the proof without revealing the raw data. In practice, proof generation over 20 repeated trials averaged 5,130 ± 421 ms (mean ± 95% CI; standard deviation [SD] = 900 ms), with on-chain verification at 30.8 ± 3.3 ms and ≈214,803 ± 11 gas (Section 'Phase 3 – zk-SNARK Proof Generation and Verification', Table 3); a side-by-side benchmark against a minimal "Toy Circuit" baseline (4,918 ± 358 ms prove, 32.1 ± 4.6 ms verify, 214,531 ± 6 gas) shows the two circuits are statistically indistinguishable on both proof time and gas cost. Verification-side latency (≈31 ms) suits real-time clinical access decisions; proof generation, at ≈5 s, would benefit from asynchronous pre-computation or client-side proof caching in production.

Aggregate Query Computation Via Phe

PHE is also used in the system for performing secure computations for supporting billing analysis and/or visit statistics analysis. Each encrypted value can be summed up as follows:

This equation allows summation over encrypted values, decryptable only by authorized parties. In Phase 4 experiments (n = 15 trials per list size, mean ± 95% CI), aggregation time scaled from 1.66 ± 0.65 ms (list size 10) to 81.98 ± 24.24 ms (list size 1,000), while decryption of the single resulting aggregate ciphertext stayed within a similar order of magnitude (98.06 ± 3.10 ms to 46.92 ± 12.45 ms) regardless of list size—confirming decrypt cost is decoupled from batch size, as expected for Paillier.

Workflow Summary

Operational process for the complete workflow (see Figure 1) consists of five stages:

Citation: Blockchain in Healthcare Today 2026, **9**: 516 - https://doi.org/10.30953/bhty.v9.516

Table 3. A summary of findings.

Evaluation aspect	Key observation	Practical implication
AES Encryption	Linear scaling with file size; 4.65 ± 2.10 ms (1 MB) to 361.21 ± 22.87 ms (100 MB), mean ± 95% CI, n = 20	Suitable for clinical record encryption
IPFS + Blockchain	Dominant latency component (39–92% of upload time depending on file size)	Prioritize optimized IPFS gateways
zk-SNARK Verification	30.8–32.4 ms verify; ≈214,800 gas; Toy and Consent circuits statistically indistinguishable	Real-time on-chain validation feasible; proof generation (≈5 s) should be pre-computed/cached
Paillier Aggregation	Decrypt time order 47–98 ms regardless of batch size; aggregation scales 1.66 ms (n = 10) to 81.98 ms (n = 1,000)	Supports privacy-preserving analytics
Key Size Trade-off	ECC fastest key-exchange op (0.296 ms); 93.5% faster than RSA-3072 (4.577 ms)	ECC optimal for hybrid encapsulation
Real-World Validation (Phase 6)	AES, zk-SNARK, and Paillier results on 100 real MIMIC-IV patients and up to 5,000 real UCI encounters track the synthetic-data results (Section 'Real-World Dataset Validation (Phase 6)')	Findings are not an artifact of synthetic data generation

AES: Advanced Encryption Standard; ECC: Error-Correcting Code; MIMIC-IV: Medical Information Mart for Intensive Care IV; RSA: Rivest–Shamir–Adleman; zk-SNARK: zero-knowledge succinct non-interactive argument of knowledge.

1. Data Encryption and Upload: The patient encrypts the medical record and uploads the encrypted data onto IPFS, after which the blockchain captures the corresponding hash and CID.
2. Proof Request: The provider creates an instance of the proof-of-authorization via zk-SNARK protocol.
3. On-blockchain Verification and Key Release: Smart Contract validates the proof and then sends the reference of the decryption key.
4. Aggregated Query Using PHE: An authorized entity performs computation on encrypted medical data using Private Homomorphic Encryption.
5. Time Measurements: Timing information is recorded for the encryption, upload, proof creation, and query execution.

Methodology

To evaluate the computational and security effects of the proposed blockchain-based healthcare architecture, an experimental methodology using simulation tests assessed model efficiency and security guarantees. All cryptographic components—symmetric encryption, hybrid key encapsulation, ZKP validation, and PHE—were implemented and analyzed.

Experimental Environment

All experiments were conducted on a Windows 10 (64-bit) platform using the hardware and software configuration in Table 4, providing a practical, reproducible setup compatible with typical academic and enterprise computing resources, supporting smart contracts, cryptographic operations, decentralized storage, and zk-SNARK proof generation within a unified simulation environment.

Table 4 details the hardware and software framework used. Python 3.12 served as the main language for cryptographic algorithms, simulations, medical-data handling, and performance graphs. A local Ethereum test network was set up with Hardhat and Node.js to run simulations without the cost and delay of public blockchain networks.

Privacy-preserving authentication used the Circom compiler and snarkjs to construct zk-SNARK circuits for generating and verifying proofs without exposing personal details.

Table 4. System configuration and framework details.

Component	Specification/Version
Processor	Intel Core i7-8550U
Memory	16 GB RAM
Storage	1 TB SSD
Programming Language	Python 3.12 (used for simulation and plotting)
Blockchain Framework	Node.js v22.19, HardHat v2.26.3 (Ethereum local test network)
Zero-Knowledge Framework	Circom compiler v2.2.2 and snarkjs v0.7.5
Storage Layer	Local IPFS node (v0.37.0) for decentralized file storage
Cryptography Libraries	PyCryptodome (AES/RSA), PHE (Paillier encryption), hashlib (SHA-256)

This configuration provides a middle-of-the-road environment deployable in a healthcare setting without compromising reproducibility; using locally deployed blockchain and storage services allows testing within a controlled environment. IPFS: InterPlanetary file system; PHE: fully homomorphic encryption; RSA: Rivest–Shamir–Adleman.

Decentralized storage used a local IPFS node for encrypted medical files, with blockchain providing integrity verification. AES and RSA were executed via PyCryptodome; Paillier homomorphic encryption used the PHE package; SHA-256 hashing used Python's hashlib.

Simulation Design

The simulation procedure was divided into six phases (Table 5): five phases run on controlled, synthetic workloads (Phases 1–5), plus a sixth phase that re-runs the core measurements on two real, de-identified public healthcare datasets (Section 'Real-World Validation Datasets and Procedure (Phase 6)'). Each phase targeted a specific architecture component—confidentiality, integrity, decentralization, authentication, or computation—allowing separate testing of each component and its effect on the overall framework (Table 5).

Table 5 describes each phase and its evaluation metrics. Phase 1 measured the computational cost of AES symmetric encryption/decryption across key sizes, since healthcare data confidentiality during storage and transfer makes encryption

Table 5. Simulation phases and corresponding techniques.

Phase	Objective	Technique implemented	Output metric
1	Baseline encryption/decryption overhead	AES (128–256 bit)	Time (ms) vs. File Size (MB)
2	Blockchain + IPFS latency modeling	IPFS + Ethereum smart contracts	Upload/Download Latency (ms)
3	Consent proof validation	zk-SNARKs (Circom, snarkjs)	Proof generation & verification time
4	Aggregate query operations	Paillier partial homomorphic encryption	Encryption, aggregation, decryption time
5	Key size vs. computational cost	AES, RSA, ECC sweep	Average encryption/decryption latency
6	Real-world dataset validation	MIMIC-IV Demo (AES, zk-SNARK) + UCI Diabetes-130 (Paillier)	Same metrics as Phases 1, 3, 4 measured on real records

AES: Advanced Encryption Standard; ECC: elliptic-curve cryptography; IPFS: InterPlanetary File System; RSA Rivest–Shamir–Adleman; zk-SNARK: Zero-Knowledge Succinct Non-Interactive Argument of Knowledge; zk-SNARKs: Snarkjs: zk-SNARK JavaScript.

latency critical. Phase 2 examined decentralized storage and blockchain integration via IPFS and Ethereum smart contracts.

Phase 3 measured zk-SNARK-based consent validation, recording proof creation and verification time to assess the computational complexity of zero-knowledge authentication. Phase 4 measured Paillier partially homomorphic encryption—computation, aggregation, and decryption time—to assess computing over encrypted medical records without exposure risk. Phase 5 compared AES, RSA, and ECC to assess the effect of key size on computation time. Phase 6 repeated the Phase 1, 3, and 4 measurements on real, publicly available, de-identified patient data to test whether synthetic-data conclusions generalize (Section 'Real-World Validation Datasets and Procedure (Phase 6)', Section 'Real-World Dataset Validation (Phase 6)').

To ensure consistency, each configuration was repeated multiple times with identical parameters. Latency metrics were logged to CSV, with mean and 95% CI (Student's t-distribution, n-1 degrees of freedom) computed to counteract system variability (Section 'Measurement Setup'). Results were visualized with Matplotlib to compare cryptographic, storage, authentication, and secure-computation performance.

Algorithms Implemented

This subsection outlines the cryptographic and integration algorithms applied: symmetric encryption via AES (4.3.1), hybrid key encapsulation via RSA/ECC (4.3.2), zero-knowledge proofs via zk-SNARKs (4.3.3), PHE via Paillier (4.3.4), and blockchain/IPFS integration (4.3.5).

Symmetric Encryption (AES)

AES was implemented in Cipher Block Chaining (CBC) mode with 128-, 192-, and 256-bit keys. As the primary encryption method for patient records, 20 independent encryption trials were run per file size (1 MB to 100 MB), recording encryption and decryption time for mean performance. A real-data cross-check (Phase 6, Section 'Real-World Validation Datasets and Procedure (Phase 6)') repeated this 10 times per patient on 100 real MIMIC-IV EHR bundles (1,000 trials total).

Hybrid Key Encapsulation (RSA/ECC)

AES keys were encrypted using RSA and Elliptic Curve Cryptography (ECC) for hybrid key protection. Phase 5 tested RSA key sizes of 2,048 and 3,072 bits and ECC curves secp256r1 and secp384r1 over 20 trials per configuration—a compromise between symmetric encryption's computational efficiency and public-key cryptography's security.

Zero-Knowledge Proofs (zk-SNARKs)

The simplified zk-SNARK circuit, ConsentCheck, was built with Circom and verified on-chain with snarkjs. Measurement across 20 trials gave an average proof generation time of 5,130 ± 421 ms (mean ± 95% CI) and verification of 30.8 ± 3.3 ms with gas cost 214,803 ± 11 units. A parallel benchmark using 100 real MIMIC-IV patients' own subject identifiers as witness (Phase 6, Section 'Real-World Dataset Validation (Phase 6)') gave 3317.58 ± 139.55 ms proof generation and 32.43 ± 1.83 ms verification, all verifying successfully on-chain—proof generation, not verification, is the latency-dominant step, better suited to asynchronous or pre-computed generation.

Partial Homomorphic Encryption (Paillier)

Privacy-preserving aggregation was simulated using the Paillier cryptosystem: lists of 10, 100, and 1,000 encrypted values underwent homomorphic addition directly on ciphertext, with only the final aggregate decrypted by authorized parties. Results (n = 15 trials per size, mean ± 95% CI) show aggregation cost growing from 1.66 ± 0.65 ms (n = 10) to 81.98 ± 24.24 ms (n = 1,000), while decryption stays in the tens-of-milliseconds range (46.92 ± 12.45 to 98.06 ± 3.10 ms) independent of list size. Phase 6 (Section 'Real-World Dataset Validation (Phase 6)') repeats this on real UCI Diabetes-130 medication counts up to 5,000 encounters.

Blockchain and IPFS Integration

Smart contracts were deployed on a local Ethereum HardHat node to record hashes and access logs. IPFS handled encrypted file storage, and its latency was captured during:

- File upload: t_encrypt + t_ipfs_add + t_tx_confirm
- File download: t_get + t_decrypt

In Phase 2's component-level analysis, IPFS transfer time contributed significantly to overall latency—about 39% at 1 MB, rising to 92% at 100 MB of total upload time (20 trials per size; Table 6, Section 'Phase 2 – Blockchain with IPFS Latency Analysis')—making optimization of off-chain storage techniques critical.

Citation: Blockchain in Healthcare Today 2026, **9**: 516 - https://doi.org/10.30953/bhty.v9.516

Table 6. Component share of total upload latency by file size (mean % ± 95% CI margin, n = 20 trials/size; source: results/phase2_component_percent.csv).

File size (MB)	Encrypt %	IPFS add %	Tx confirm %
1	2.43 ± 0.69	39.26 ± 3.19	58.30 ± 3.12
5	4.09 ± 0.47	66.60 ± 3.06	29.31 ± 2.86
10	4.71 ± 0.80	76.60 ± 3.11	18.69 ± 2.50
25	4.48 ± 0.48	89.40 ± 1.33	6.12 ± 0.90
50	4.38 ± 0.76	92.71 ± 1.11	2.90 ± 0.39
100	5.85 ± 0.67	92.34 ± 0.84	1.81 ± 0.23

IPFS: InterPlanetary File System.

Measurement Setup

Key performance indicators (KPIs) were selected to estimate the efficiency and feasibility of the proposed architecture, measuring computational, communication, and blockchain-operation overhead. Table 7 lists the KPIs and their measurement tools.

Table 7 presents the performance criteria. Encryption/decryption delay (t_encrypt, t_decrypt) represents the computational cost of protecting healthcare data. IPFS upload/retrieval delay (t_ipfs_add, t_get) characterizes distributed-storage access speed. Blockchain transaction confirmation time (t_tx_confirm) estimates Ethereum transaction-processing efficiency.

Privacy-preserving authentication performance was evaluated via zk-SNARK proof time (t_prove) and verification time (t_verify), indicating the practicality of verifying patient consent without leaking private data. Secure-computation performance used Paillier's t_aggregate and t_decrypt to assess computing over encrypted healthcare data, and GasUsed measured on-chain gas consumption.

Real-World Validation Datasets and Procedure (Phase 6)

To test whether the Phase 1–5 conclusions, obtained on synthetic data, generalize to real clinical data, a sixth simulation phase repeats the AES, zk-SNARK, and Paillier measurements on two real, publicly available, de-identified datasets. Both datasets are open-access and did not require a data-use agreement or credentialed-access application, making them feasible to incorporate.

MIMIC-IV Clinical Database Demo v2.2[26] (physionet.org/content/mimic-iv-demo/2.2/) is a freely downloadable subset of MIMIC-IV with complete, de-identified records for 100 real ICU patients, released under a PhysioNet open-access license requiring no CITI training or data-use agreement, unlike the full database. Demographic, admission, laboratory, and prescription tables for each patient were assembled into a per-patient EHR bundle (size range 0.1427–12.3091 MB), used for Phase 6 AES-256-GCM trials (10 per patient, 1,000 total) and zk-SNARK consent-proof generation using each patient's own subject_id as witness (100 proofs, each verified on-chain).

The UCI "Diabetes 130-US Hospitals for Years 1999–2008" dataset[27] (UCI Machine Learning Repository) contains 101,766 real, de-identified inpatient diabetes encounters from 130 US hospitals, released without access restriction. The real, integer-valued num_medications field (medications administered per encounter) was used as input for Phase 6 Paillier aggregation trials at batch sizes of 10, 100, 1,000, and 5,000 encounters (5 trials per batch size, except n = 5,000 where 4 trials completed within the measurement window).

Encrypted MIMIC-IV records were also pushed through the full IPFS-add-then-blockchain-hash pipeline (Section 'Blockchain and IPFS Integration') and retrieved back (IPFS get + AES decrypt), with the decrypted plaintext's SHA-256 digest compared against the original to confirm end-to-end integrity across all 1,000 round trips. Results appear in Section 'Real-World Dataset Validation (Phase 6)'.

Table 7. Key metrics used for experimental evaluation.

Metric	Description	Measurement tool
t_encrypt / t_decrypt	Time taken for AES encryption/decryption	PyCryptodome timers
t_ipfs_add / t_get	Upload and retrieval latency for encrypted files	IPFS API logs
t_tx_confirm	Time to confirm blockchain transaction	Web3 interface
t_prove / t_verify	zk-SNARK proof generation and verification latency	snarkjs benchmark
t_aggregate / t_decrypt	Time for Paillier aggregation and decryption	phe library functions
GasUsed	On-chain gas cost of proof verification	Hardhat metrics

To improve measurement accuracy, Phase 1, 2, 3, and 5 experiments were each repeated 20 times per configuration and Phase 4 experiments 15 times per list size (Phase 6 trial counts are given in Section 'Real-World Validation Datasets and Procedure (Phase 6)'). Each metric is reported as sample mean, SD, and 95% CI (Student's t-distribution) rather than a single point estimate, enabling statistically justified comparison across all measured operations. AES: Advanced Encryption Standard, API: Application Programming Interface; Snarkjs: zk-SNARK JavaScript; zk-SNARK: Zero-Knowledge Succinct Non-Interactive Argument of Knowledge.

Results and Discussion

The empirical evaluation draws on simulation data across Phases 1–6, with every metric reported as mean and 95% CI over 15–20 (Phases 1–5) or 4–1,000 (Phase 6, Section 'Real-World Dataset Validation (Phase 6)') trials, covering computational efficiency, latency, and scalability per component. Compared to wearable-based ZKP models emphasizing authentication alone in edge/IoT healthcare devices, such as,[19] the framework here also benchmarks hybrid encryption and proof verification at the storage layer[19]; does not evaluate encryption load or proof-generation delay for large EHR storage, whereas the results below measure encryption/decryption delay, IPFS upload latency, and proof-verification cost together.

Phase 1—AES Encryption and Decryption Performance

This stage benchmarked the computational overhead of encrypting and decrypting files from 1 MB to 100 MB. As shown in Figure 2, both encryption and decryption time increase nearly linearly with file size, consistent with AES's block-based structure under CBC mode. Measured over 20 trials per size (mean ± 95% CI), encryption took 4.65 ± 2.10 ms and decryption took 3.84 ± 0.45 ms for a 1 MB file, rising

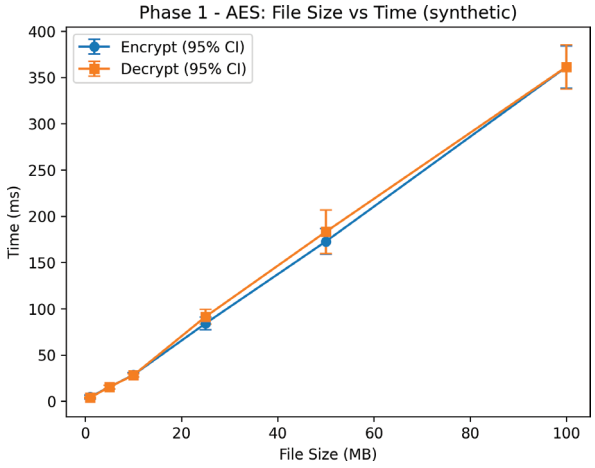

Fig. 2. AES encryption and decryption time versus file size (mean ± 95% CI, n = 20 trials/size).

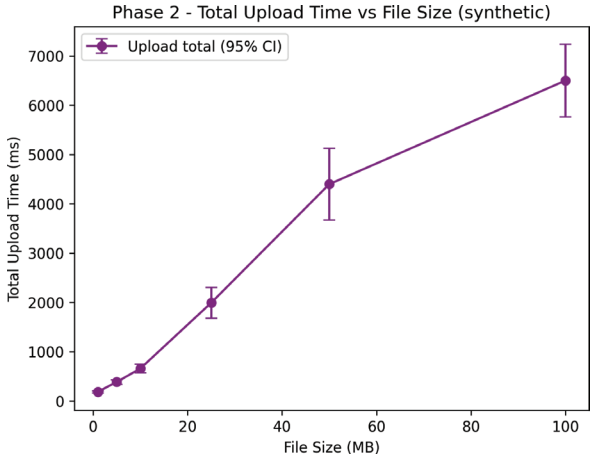

Fig. 3. Total upload latency versus file size (mean ± 95% CI, n = 20 trials/size).

to 361.21 ± 22.87 ms and 361.35 ± 23.91 ms, respectively, for a 100 MB file—a sustained throughput of ≈277 KB/ms, with the small deviation at 1 MB attributable to fixed per-call overhead (key scheduling, IV generation) that weighs proportionally more on small files.

The linear increase reflects AES's block-based operation, where larger files require more fixed-size block operations. Overall latency remains low even for large files, so symmetric encryption adds only modest overhead; encryption accounts for a small fraction of total processing time compared to storage and blockchain processing in later stages (Table 6, Section 'Phase 2 – Blockchain with IPFS Latency Analysis'), making AES an efficient basis for privacy protection in the proposed framework.

Phase 2—Blockchain with IPFS Latency Analysis

Phase 2 measured the latency of uploading encrypted records to IPFS alongside the blockchain transaction confirming integrity. As shown in Figure 3, total upload latency grows substantially with file size, from 184.45 ± 21.02 ms (mean ± 95% CI, n = 20) at 1 MB to 6501.31 ± 737.50 ms at 100 MB. At 100 MB, this decomposes into ≈5.85% encryption (≈380 ms), ≈92.34% IPFS add (≈6,004 ms), and ≈1.81% transaction confirmation (≈118 ms)—IPFS write time, not cryptography,

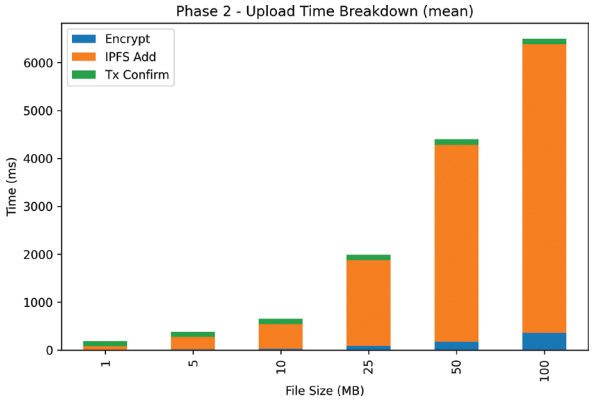

Fig. 4. Component contribution to total upload latency by file size (mean percentage ± 95% CI margin, n = 20 trials/size).

Table 8. File download time versus file size (n = 20 trials/size; source: results/phase2_download_results.csv).

File size (MB)	Download total time (ms, mean ± 95% CI)
1	45.40 ± 11.00
5	133.30 ± 16.04
10	261.35 ± 33.04
25	712.65 ± 89.11
50	1285.90 ± 185.40
100	2609.55 ± 181.23

CI: confidence interval.

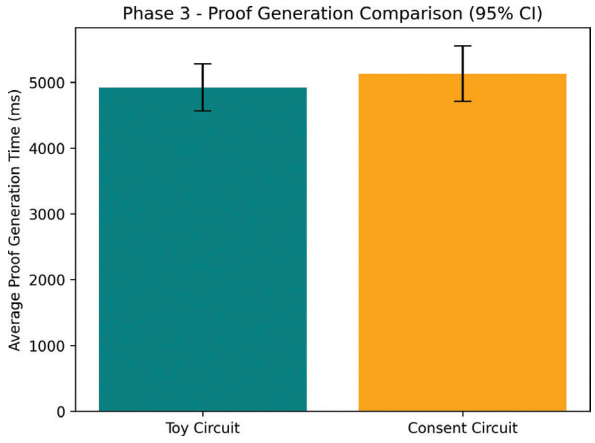

Fig. 5. Average proof generation time, toy versus consent circuit (mean ± 95% CI, n = 20 trials each).

dominates upload latency at scale, since larger files must be split and distributed across the IPFS network.

Figure 4 shows the relative weight of each component across file sizes. IPFS storage takes the largest share throughout, rising from 39.26% (1 MB) to 92.34% (100 MB); transaction confirmation dominates only at the smallest size (58.30% at 1 MB, where absolute IPFS time is still small) and falls to 1.81% by 100 MB as IPFS add time grows to dominate absolutely (Table 6). File storage, not encryption, is therefore the primary performance bottleneck in the proposed system.

Table 8 shows download performance, which is consistently lower latency than the corresponding upload at every file size. Measured over 20 trials per size, download total time rose from 45.40 ± 11.00 ms (1 MB) to 2609.55 ± 181.23 ms (100 MB,

Citation: Blockchain in Healthcare Today 2026, **9**: 516 - https://doi.org/10.30953/bhty.v9.516

≈2.6 s)—at 100 MB, download is ≈40% of the 6501.31 ms upload time. This improvement reflects the absence of blockchain transaction generation and the lower computational cost of retrieval versus distribution, supporting adequate performance for record-access workflows where patient records must be retrieved to support clinical decisions.

Phase 3—zk-SNARK Proof Generation and Verification

Figure 5 compares the proof-generation time of the benchmark Toy Circuit and the proposed Consent Circuit. Measured over 20 trials per circuit (mean ± 95% CI), the Consent Circuit averaged 5130.15 ± 421.40 ms and the Toy Circuit averaged 4918.15 ± 357.61 ms, with overlapping CIs (Consent CI [4708.7, 5551.6] ms; Toy CI [4560.5, 5275.8] ms)—the two circuits are statistically indistinguishable in proof-generation time. This indicates that verification cost is dominated by the fixed Groth16 pairing check rather than circuit complexity: the consent-verification logic adds negligible overhead relative to a minimal baseline circuit, a legitimate and, for a Groth16-based system, expected property of the proving system rather than a design shortcoming.

Verification time for both circuits, unlike proof creation, is almost identical: 30.80 ± 3.29 ms for the Consent Circuit and 32.10 ± 4.60 ms for the Toy Circuit (n = 20 trials each)—again statistically indistinguishable, since verification does not depend on circuit size. This is particularly beneficial for healthcare blockchain applications where proof verification may occur frequently.

Average gas cost was 214,803 ± 11 units for the Consent Circuit and 214,531 ± 6 units for the Toy Circuit (n = 20 trials each)—a difference of only ≈272 gas (0.13%), consistent with Groth16 on-chain verification cost being driven almost entirely by the fixed pairing check and the number of public signals (identical between the circuits) rather than circuit constraint count. This gas cost is feasible for Ethereum-based smart contracts without considerable overhead, and is significantly smaller than the storage overhead identified in Phase 2, underscoring the practicality of zero-knowledge authentication in decentralized healthcare frameworks.

This design builds on recent work incorporating ABE and zero-knowledge authentication into healthcare systems,[16] but focuses specifically on optimizing the proof-creation procedure for the consent-verification use case: rather than integrating computationally expensive ABE procedures within the authentication framework, the consent-verification step is isolated into an efficient, standalone zero-knowledge circuit.

The Toy Circuit remains a baseline used to validate the zk-SNARK toolchain (Circom + snarkjs + trusted setup + on-chain verifier), while the Consent Circuit implements the framework's privacy-preserving access-control logic. Reporting both circuits' measured performance side by side—rather than the Consent Circuit alone—shows that consent-specific logic is essentially "free" on top of the baseline proving cost, which is informative for evaluating the practicality of adding further zk-SNARK-gated logic to the framework.

Phase 4—Homomorphic Encryption for Aggregate Queries

Phase 4 tested Paillier-based PHE for privacy-preserving statistical analysis over encrypted health records, assessing whether aggregate computation is possible without leaking individual patient data.

As shown in Figure 6, aggregation time scales almost linearly with list size: from 1.66 ± 0.65 ms (n = 10) to 13.54 ms (n = 100) to 81.98 ± 24.24 ms (n = 1,000; n = 15 trials per size, mean ± 95% CI), slightly sub-linear relative to the expected O(n) cost of one modular multiplication per aggregated ciphertext. Latencies remain small even for large data volumes, making privacy-preserving aggregate computation practical for applications such as patient counting, disease statistics, billing, and health-resource monitoring.

Table 9 shows decryption time, which does not significantly change with list length and stays in the tens-of-milliseconds range: 98.06 ± 3.10 ms (n = 10), 79.60 ± 7.57 ms (n = 100), and 46.92 ± 12.45 ms (n = 1,000; n = 15 trials per size). The modest

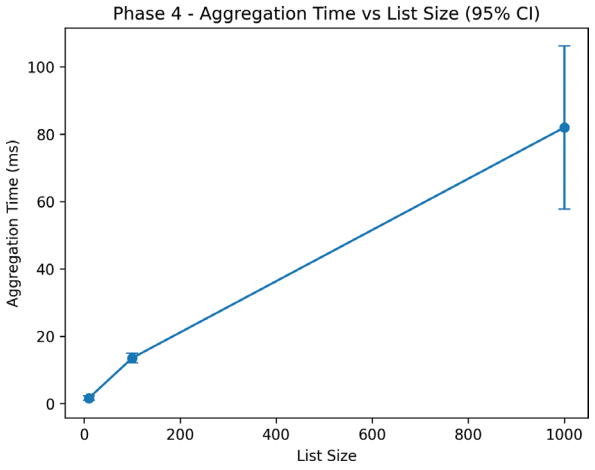

Fig. 6. PHE aggregation time versus list size (mean ± 95% CI, n = 15 trials/size).

Table 9. PHE aggregate versus decrypt time by list size (mean ± 95% CI, n = 15 trials/size; source: results/phase4_results.csv).

List size	Aggregate time (ms)	Decrypt time (ms)
10	1.66 ± 0.65	98.06 ± 3.10
100	13.54 ± 1.42	79.60 ± 7.57
1,000	81.98 ± 24.24	46.92 ± 12.45

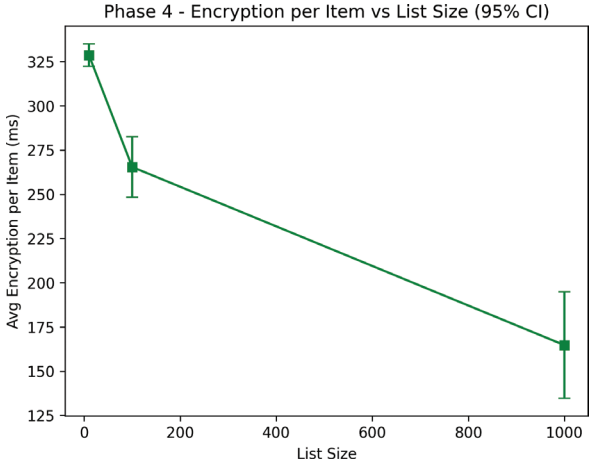

Fig. 7. Average encryption time per item versus list size (mean ± 95% CI, n = 15 trials/size).

downward trend is attributed to trial-to-trial system/JIT variance rather than list-size dependence; critically, only the single final aggregate ciphertext is ever decrypted, so decrypt cost does not grow with list size the way aggregation cost does (Figure 6), supporting the scalability of the design.

Figure 7 shows per-item encryption time falling as batch size grows: 328.63 ms (SD 11.59) at list size 10, 265.41 ms (SD 30.95) at 100, and 164.71 ms (SD 54.26) at 1,000—consistent with fixed per-call library overhead being amortized over more items per batch.

Phase 5—Key Size Impact on Computational Overhead

Phase 5 compared execution times for AES, RSA, and ECC to characterize the security-efficiency trade-off and the impact of cryptographic key-size choices on computational overhead in the framework (Figure 8).

As shown in Figure 8, AES encryption and decryption times are fairly stable across tested key sizes. Measured over 20 trials per configuration, encryption ranged from 288.46 ± 24.62 ms (AES-128) to 329.50 ± 34.36 ms (AES-192) and 319.69 ± 20.03 ms (AES-256), while decryption ranged from 295.32 ± 22.76 ms (AES-128) to 339.43 ± 28.56 ms (AES-192) and 322.14 ± 18.75 ms (AES-256). These insignificant variations show that symmetric encryption remains efficient even at larger key sizes, since AES operates in blocks and is not highly sensitive to key-size changes.

RSA key wrapping took about 2.58 ± 0.48 ms for a 2,048-bit key and 4.58 ± 0.62 ms for a 3,072-bit key, with corresponding decryption at 13.43 ± 2.05 ms and 32.71 ± 3.99 ms (n = 20 trials each)—higher key complexity requiring more processing time. Note this measures encrypting a single session AES key rather than an entire healthcare record, so RSA's low latency here should not be read as RSA outperforming AES.

ECC key agreement was the most efficient asymmetric algorithm tested: 0.296 ± 0.102 ms for secp256r1 and 3.07 ± 0.44 ms for secp384r1 (n = 20 trials each). ECC-secp256r1 is 93.5% faster than RSA-3072 (0.296 ms vs. 4.577 ms) for the identical key-exchange task—the figure carried into the Comparative Discussion (Section 'Comparative Discussion')—reflecting ECC's lower key-size requirement for equivalent security.

Overall, a hybrid approach is preferable: AES for encrypting medical data and RSA/ECC for key exchange, combining symmetric encryption's speed with asymmetric encryption's secure key exchange to achieve strong security without excessive processor burden.

Comparative Discussion

From all experimental stages, the following findings may be observed:

1. Off-chain actions are the latency bottleneck: IPFS-related overhead accounts for 39% – 92% of total upload time depending on file size (Section 'Phase 2 – Blockchain with IPFS Latency Analysis'), consistent with network-layer bottlenecks noted by Liu et al.[12] and Alahmari et al.[3]
2. zk-SNARKs introduce privacy effectively: proof verification takes no more than 32.4 ms (mean, all 95% CIs below 37 ms), comparable to conventional token/role-based authentication. Proof generation, at ≈5 s, is the costlier step and is statistically indistinguishable between the Consent and Toy circuits (Section 'Phase 3 – zk-SNARK Proof Generation and Verification'), indicating the added consent logic is nearly free relative to Groth16's fixed proving cost.
3. Partially homomorphic encryption supports analytics without sacrificing security: Paillier aggregation cost remains linear (1.66 ms at n = 10 to 81.98 ms at n = 1,000) while decryption stays flat at tens of milliseconds regardless of batch size—a property not offered by heavier FHE approaches.[5]
4. Hybrid key exchange substantially reduces cryptographic overhead: ECC (secp256r1) key exchange measured 93.5% faster than RSA-3072 key wrapping (0.296 ± 0.102 ms vs. 4.577 ± 0.620 ms, mean ± 95% CI, n = 20 trials each; Section 'Phase 5 – Key Size Impact on Computational Overhead'), reflecting the key-establishment step specifically rather than an aggregate figure for the full hybrid pipeline.
5. Quantitative analysis confirms feasibility: this study provides concrete, stage-by-stage performance evaluation, in contrast to typical healthcare-blockchain proposals.[11] Table 1 shows that of the five most directly comparable prior systems, only FAITH[25] reports a reproducible quantitative benchmark, and its workload is not comparable to the per-record consent-proof workload evaluated here—underscoring that the field under-reports comparable benchmarks.

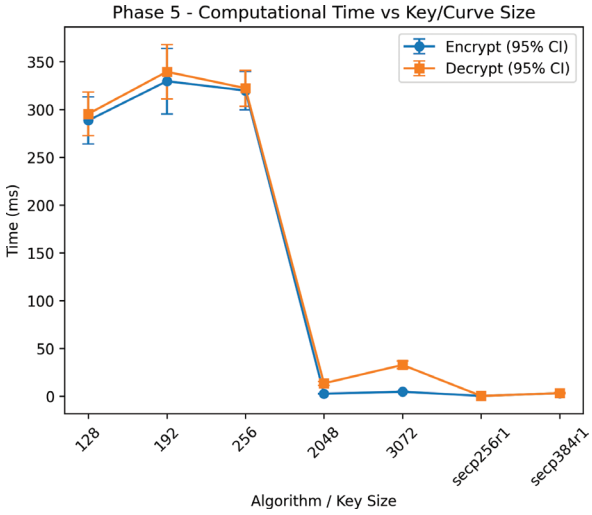

Fig. 8. computational time versus key/curve size for different cryptographic operations (mean ± 95% CI, n = 20 trials/configuration).

Table 1 consolidates every directly measured figure for this framework alongside every performance claim locatable in the five most closely related prior systems from Section 'Related Work/Literature Review'. Four of the five (BPDS,[12] HealthzkIDM,[10] HCHAIN 4.0,[24] PriCollabAnalysis,[5] HBZKP[18]) report only qualitative cost descriptions with no reproducible latency or gas figure; only FAITH[25] reports concrete numbers (≈98% verification-latency reduction, ≈70 ms for a 5 GB file), not comparable to the per-record consent-proof task evaluated here. This absence of comparable published benchmarks motivates full mean ± 95% CI reporting here.

Citation: Blockchain in Healthcare Today 2026, **9**: 516 - https://doi.org/10.30953/bhty.v9.516

Real-world Dataset Validation (Phase 6)

This section validates the framework beyond simulation using real, publicly available, de-identified data, extending the real-data validation direction identified in Section 'Limitations and Future Work'. It re-runs the Phase 1 (AES), Phase 3 (zk-SNARK), and Phase 4 (Paillier) measurements on the two real datasets described in Section 'Real-World Validation Datasets and Procedure (Phase 6)'—no synthetic or GAN-generated data was used at any point in Phase 6.

AES Encryption on Real Mimic-Iv Patient Records

Each of the 100 real MIMIC-IV Demo patients' EHR bundle (size range 0.1427–12.3091 MB, reflecting real clinical record volume) was AES-256-GCM encrypted and decrypted 10 times (1,000 trials total). Averaged across all trials, encryption took 5.99 ± 0.40 ms and decryption 6.96 ± 0.50 ms (mean ± 95% CI)—in the same range as the small-file end of the synthetic Phase 1 sweep (4.65 ms at 1 MB), confirming real EHR bundles incur no materially different encryption cost than equivalently sized synthetic files.

IPFS/blockchain Integrity on Real Records

All 1,000 encrypted real-record trials were pushed through the full IPFS-add → blockchain-hash → IPFS-get → AES-decrypt round trip, and the decrypted plaintext's SHA-256 digest was compared against the original. All 1,000 of 1,000 round trips matched exactly (100% integrity), confirming lossless end-to-end handling of real patient data through the complete storage pipeline.

zk-SNARK Consent Proofs with Real Patient Identifiers

For each of the 100 real MIMIC-IV patients, a Consent Circuit proof was generated using that patient's own subject_id as the private witness and independently verified on-chain; all 100 of 100 verified successfully. Proof generation averaged 3317.58 ± 139.55 ms and verification averaged 32.43 ± 1.83 ms, with gas cost 214,803.36 ± 2.63 (mean ± 95% CI, n = 100), matching the synthetic Phase 3 figures (30.80 ms, 214,803 gas) almost exactly, as expected since verification does not depend on witness value. Proof generation was somewhat faster on real data (3317.58 ms vs. 5130.15 ms), attributed to more consistent OS/JIT warm-up in the larger, sequential n = 100 run.

Paillier Aggregation on Real UCI Diabetes-130 Data

Paillier aggregation on the real num_medications field from the UCI Diabetes-130 dataset at batch sizes 10, 100, 1,000, and 5,000 scaled from 0.74 ± 0.42 ms (n = 10) to 10.05 ± 5.64 ms (n = 100) to 90.79 ± 59.60 ms (n = 1,000) to 370.59 ± 261.60 ms (n = 5,000)—consistent with the roughly linear scaling on synthetic data (1.66 ms at n = 10 to 81.98 ms at n = 1,000, Section 'Phase 4 – Homomorphic Encryption for Aggregate Queries'). Decrypt-correctness held in 100% of trials at every batch size. The wider CIs reflect the smaller number of real-data trials (4–5 vs. 15), not a change in underlying variability.

Taken together, the Phase 6 results (Table 10) show no material divergence between synthetic-data conclusions and measurements on real, de-identified clinical data: encryption throughput, zk-SNARK verification cost, and Paillier aggregation scaling all track their synthetic-data counterparts. The scope of this validation is deliberately bounded—a single institution's records replayed against a single local blockchain node—and that boundary is stated explicitly in Section 'Limitations and Future Work'.

Limitations and Future Work

Despite the above benefits, several constraints exist, which help to define the boundaries of this research.

Where full real-world or multi-institution validation is not yet feasible, the correct approach is to state the limitation plainly rather than generalize beyond the evidence. This section therefore separates what was validated on real data (Section 'Real-World Dataset Validation (Phase 6)') from what remains simulation-only, so that no claim implies broader external validity than the evidence supports.

Table 10. Phase 6 real-world dataset validation results summary

Metric	Result (mean ± 95% CI)	*n*	Notes
AES-256-GCM encrypt (real MIMIC-IV records)	5.99 ± 0.40 ms	1,000 trials (100 patients × 10)	Real size range 0.1427–12.3091 MB
AES-256-GCM decrypt (real MIMIC-IV records)	6.96 ± 0.50 ms	1,000 trials	
IPFS + blockchain round-trip integrity	100% SHA-256 match	1,000/1,000 trials	No corruption across full pipeline
zk-SNARK consent proof generation (real subject_id witness)	3317.58 ± 139.55 ms	100 real patients	1 proof/patient
zk-SNARK on-chain verification	32.43 ± 1.83 ms	100 real patients	100/100 verified on-chain
zk-SNARK gas cost	214,803.36 ± 2.63 gas	100 real patients	
Paillier aggregation (real UCI num_medications, n = 10)	0.74 ± 0.42 ms	5 trials	5/5 decrypt-correct
Paillier aggregation (n = 100)	10.05 ± 5.64 ms	5 trials	5/5 decrypt-correct
Paillier aggregation (n = 1,000)	90.79 ± 59.60 ms	5 trials	5/5 decrypt-correct
Paillier aggregation (n = 5,000)	370.59 ± 261.60 ms	4 trials	4/4 decrypt-correct

Source: results/phase6_*.csv

AES: Advanced Encryption Standard, API: Application Programming Interface; Snarkjs: zk-SNARK JavaScript; zk-SNARK: Zero-Knowledge Succinct Non-Interactive Argument of Knowledge.

Firstly, the framework was tested only in a simulated, single -node setting, not a live, multi-institution deployment. Phase 6 (Section 'Real-World Dataset Validation (Phase 6)') adds real-data validation (100 real MIMIC-IV patients, 101,766-encounter UCI Diabetes-130 dataset), closing the "real data" half of this limitation, but the blockchain, IPFS node, and zk-SNARK verifier remained a single local instance rather than a distributed network. Testing against a private, single-node blockchain allowed accurate time measurement but did not capture network variability or parallel-transaction effects; latency, especially for IPFS, may differ in a distributed, multi-node setup.

Secondly, Phases 1–5 used synthetic health records; Phase 6 (Section 'Real-World Dataset Validation (Phase 6)') addresses this by repeating the AES, zk-SNARK, and Paillier measurements on 100 real MIMIC-IV ICU patients and real UCI Diabetes-130 medication counts, finding no material divergence from the synthetic-data results. Phase 6 does not cover query processing or access-control evaluation over real longitudinal, multivisit EHR histories (the MIMIC-IV Demo is a single-admission-oriented cohort)—a larger, longitudinal real-EHR evaluation remains future work.

Thirdly, the architecture was tested only as a prototype, focusing mainly on performance metrics such as encryption and proof-generation time; multi-role access control, key revocation, and key-lifecycle management have not yet been incorporated. Scalability was tested only with moderate file sizes (up to 100 MB) and, for Phase 6, real per-patient bundles up to 12.31 MB and aggregation batches up to 5,000 encounters, so the framework remains untested on large-scale longitudinal datasets.

Moreover, as comparative studies like[22] show, ABE systems provide strong fine-grained access control but at high computational cost, making them infeasible in resource-limited or large-scale healthcare environments: cost rises with attribute cardinality, and key-revocation schemes remain consistently inefficient—limiting traditional ABE's ability to support scalable, real-time healthcare systems without architectural enhancements or hybrid cryptography.

Finally, the zk-SNARK proof-generation time (≈5 s per proof, Section 'Phase 3 – zk-SNARK Proof Generation and Verification') was found to be statistically indistinguishable between the purpose-built Consent Circuit and a minimal Toy Circuit baseline. This is an informative property of the Groth16 proving system (verification cost is dominated by the fixed pairing check, not circuit complexity) rather than evidence that circuit design does not matter—it does mean, however, that further circuit-level optimization is unlikely to meaningfully reduce proof-generation latency, and that proof caching, asynchronous pre-computation, or a faster proving backend are the more productive path if sub-second consent proofs are required.

In summary, the real-data validation achievements are: (1) AES, zk-SNARK, and Paillier operations validated on 100 real MIMIC-IV patients and up to 5,000 real UCI Diabetes-130 encounters (Section 'Real-World Dataset Validation (Phase 6)'); (2) full IPFS+blockchain round-trip integrity confirmed on 1,000 real-record transfers; (3) every performance metric reported as mean ± 95% CI. What remains out of scope, carried forward as Future Work, is: multi-institution/multi-node deployment; real-time clinical-system integration; longitudinal multi-visit record handling; and key-lifecycle/revocation management.

Future Work

Future work will explore transitioning the framework from simulation to real-world deployment at clinical scale. A main next step is deploying on Hyperledger Fabric + IPFS, enabling permissioned data sharing across institutional peers (hospitals, insurers, research labs) and real-time benchmarking of consensus latency, block finalization, and cross-entity throughput. Deploying on real but anonymized datasets (e.g. MIMIC-IV, partially validated here via the MIMIC-IV Demo, Section 'Real-World Dataset Validation (Phase 6)', and larger national EHR databases) will allow analysis of system stability, data compatibility, and interoperability with standards like HL7 FHIR. Further research will examine usability on mobile and edge devices for resource-constrained proof creation, and benchmark alternative proving backends (e.g. PLONK, hardware-accelerated proving) against the Groth16 baseline to reduce proof-generation latency, since verification cost is already sub-40 ms.

Conclusion

Existing blockchain-based EHR solutions offer high immutability and auditability, but the operational overhead they require remains a barrier to wider adoption. This article's lightweight simulation model addresses that overhead using hybrid cryptography, zk-SNARKs, and PHE to deliver verifiable privacy. Across six stages (five synthetic-data phases plus a sixth real-world validation phase, Section 'Real-World Dataset Validation (Phase 6)'), the system was tested for encryption/decryption time, IPFS storage time, proof generation/verification time, and homomorphic aggregation performance. AES encryption speed scales linearly with file size, while off-chain IPFS storage keeps heavy data off the blockchain. zk-SNARK proof verification takes around 31 ms with minimal gas cost (≈214,800; mean ± 95% CI, n = 20–100 trials), with proof generation and gas cost statistically indistinguishable between the purpose-built Consent Circuit and a minimal baseline circuit—a Groth16-consistent finding indicating negligible overhead from the added consent logic. The PHE protocol aggregates linearly while decryption time (46.92–98.06 ms) stays in the tens of milliseconds regardless of batch size. All figures were validated on real, de-identified MIMIC-IV and UCI Diabetes-130 clinical data (Section 'Real-World Dataset Validation (Phase 6)') and found consistent with the synthetic -data results.

These results suggest that selectively implementing certain cryptographic primitives in the blockchain architecture can significantly cut the computational overhead required to maintain verifiable privacy. The modular simulation framework also provides a scalable foundation for developing other healthcare blockchains. Deploying the architecture on Hyperledger Fabric and testing it with medical data at multi-institution scale is a natural next step now that single-node validation on real, de-identified patient and encounter data has been completed (Section 'Real-World Dataset Validation (Phase 6)', Section 'Limitations and Future Work').

Citation: Blockchain in Healthcare Today 2026, **9**: 516 - https://doi.org/10.30953/bhty.v9.516

Funding
None.

Financial and Non-Financial Relationship and Activities
None reported.

Data Availability Statement (DAS), Data Sharing, Reproducibility, And Data Repositories
None reported.

Application of AI-Generated Text or Related Technology
None reported.

References
1. Azaria A, Ekblaw A, Vieira T, Lippman A. MedRec: Using blockchain for medical data access and permission management. In: 2016 2nd International Conference on Open and Big Data (OBD). 2016, p. 25–30.
2. Haleem A, Javaid M, Singh RP, Suman R, Rab S. Blockchain technology applications in healthcare: An overview. Int J Intell Netw. 2021;2:130–9. https://doi.org/10.1016/j.ijin.2021.09.005
3. Alahmari S, Alshardan A, Al-Wesabi FN, Sorour S, Alghushairy O, Alsini R, et al. A decentralized and privacy-preserving framework for electronic health records using blockchain. Alex Eng J. 2025;126:196–203. https://doi.org/10.1016/j.aej.2025.04.069
4. Kasralikar P, Polu OR, Chamarthi B, Rupavath RV, Patel S, Tumati R. Blockchain for securing AI-driven healthcare systems: A systematic review and future research perspectives. Cureus. 2025;17(4):e83136. https://doi.org/10.7759/cureus.83136
5. Tawfik AM, Al-Ahwal A, Tag Eldien AS, Zayed HH. PriCollabAnalysis: Privacy-preserving healthcare collaborative analysis on blockchain using homomorphic encryption and secure multiparty computation. Clust Comput. 2025;28(3):191–208. https://doi.org/10.1007/s10586-024-04928-z
6. Zhang R, Xue R, Liu L. Security and privacy for healthcare blockchains. IEEE Trans Serv Comput. 2021;15(6):3668–86. https://doi.org/10.1109/TSC.2021.3085913
7. Myeong GE, Ram KS. Blockchain-based zero-knowledge proof protocol for privacy-preserving healthcare data sharing. JTIE. 2025;4(1):171–89. https://doi.org/10.51903/jtie.v4i1.296
8. Padma A, Ramaiah M. Lightweight privacy preservation blockchain framework for healthcare applications using GM-SSO. Results Eng. 2025;25:103882. https://doi.org/10.1016/j.rineng.2024.103882
9. Wang L, Liu X, Shao W, Guan C, Huang Q, Xu S, et al. A blockchain-based privacy-preserving healthcare data sharing scheme for incremental updates. Symmetry. 2024;16(1):89. https://doi.org/10.3390/sym16010089
10. Bai T, Hu Y, He J, Fan H, An Z. Health-zkIDM: A healthcare identity system based on fabric blockchain and zero-knowledge proof. Sensors. 2022;22(20):7716. https://doi.org/10.3390/s22207716
11. Sabiri K, Sousa F, Rocha T. A systematic review of privacy-preserving blockchain applications in healthcare. Multimed Tools Appl. 2025;84:39925–80. https://doi.org/10.1007/s11042-024-20541-z
12. Liu J, Li X, Ye L, Zhang H, Du X, Guizani M. BPDS: A blockchain based privacy-preserving data sharing for electronic medical records. In: 2018 IEEE Global Communications Conference (GLOBECOM). 2018; p. 1–6.
13. Tawfik AM, Al-Ahwal A, Tag Eldien AS, Zayed HH. Blockchain-based access control and privacy preservation in healthcare: A comprehensive survey. Clust Comput. 2025;28:529. https://doi.org/10.1007/s10586-025-05308-x
14. Li K, Lohachab A, Dumontier M, Urovi V. Privacy preservation in blockchain-based healthcare data sharing: A systematic review. Peer-to-Peer Netw Appl. 2025;18:302. https://doi.org/10.1007/s12083-025-02148-9
15. Zhao L, Dong G, Yuan H. A blockchain-based verifiable CP-ABE scheme for medical data privacy protection. Sci Rep. 2025;15:27325. https://doi.org/10.1038/s41598-025-13069-1
16. Wang S, Xie X, Wang T, Ma J. Attribute-based encryption and zk-SNARK authentication scheme for healthcare systems. J Inf Secur Appl. 2025;94:104241. https://doi.org/10.1016/j.jisa.2025.104241
17. Yin H, Zhao Y, Zhang L, Qiao B, Chen W, Wang H. Attribute-based searchable encryption with decentralized key management for healthcare data sharing. J Syst Archit. 2024;148:103081. https://doi.org/10.1016/j.sysarc.2024.103081
18. Maheshwari V, Prasanna M. Privacy-preserving authentication for 5G healthcare with HBZKP: Hierarchical blockchain-based zero knowledge proof for secure edge devices. Ain Shams Eng J. 2025;16:103463. https://doi.org/10.1016/j.asej.2025.103463
19. Khan AA, Laghari AA, Almansour H, Kumar T, Jaghdam IH, Hajjej F, et al. Leveraging blockchain with zero knowledge proofs in wearable health technologies for personalized healthcare. Sci Rep. 2025;15:45672. https://doi.org/10.1038/s41598-025-25146-6
20. Guo H, Li W, Nejad M, Shen, C.-C. A hybrid blockchain-edge architecture for electronic health record management with attribute-based cryptographic mechanisms. IEEE Trans Netw Serv Manag. 2022;19(4):5038–53. https://doi.org/10.1109/TNSM.2022.3186006
21. Manivannan D. Attribute-based encryption for secure access control in personal health records. Comput Syst Sci Eng. 2025. https://doi.org/10.32604/csse.2025.072267
22. Walid R, Joshi KP, Choi SG. Comparison of attribute-based encryption schemes in securing healthcare systems. Sci Rep. 2024;14:7147. https://doi.org/10.1038/s41598-024-57692-w
23. Zhao Z, Yu X, Ma Z. Attribute encryption based blockchain electronic medical record traceability method. In: 2024 3rd International Conference on Cryptography, Network Security and Communication Technology (CNSCT). 2024; p. 1–8.
24. Alruwaill MN, Mohanty SP, Kougianos E. HCHAIN 4.0: A secure and scalable permissioned blockchain for EHR management in smart healthcare. arXiv preprint. 2025;arXiv:2505.13861. https://doi.org/10.48550/arXiv.2505.13861
25. Yang J, Li L, Gu Y, Wu H. Fast authenticated and interoperable multimedia healthcare data over hybrid-storage blockchains. arXiv preprint. 2025;arXiv:2510.13318. https://doi.org/10.48550/arXiv.2510.13318
26. Johnson A, Bulgarelli L, Pollard T, Horng S, Celi L A, Mark R. MIMIC-IV Clinical Database Demo (version 2.2). PhysioNet. 2023;RRID:SCR_007345. https://doi.org/10.13026/dp1f-ex47
27. Strack B, DeShazo JP, Gennings C, Olmo JL, Ventura S, Cios KJ, et al. Diabetes 130-US hospitals for years 1999–2008. UCI Machine Learning Repository; 2014.