0, 1–10 (2025) Preprint 24 April 2025

Decentralized Healthcare Data Sharing System

Lam-Anh.Phan[1*] Thu-Minh.Tran[1] Dinh-Thuc.Nguyen [1]

[1]University of Science, VNU-HCMC, Faculty of Information System, Department of Knowledge Engineering

ABSTRACT

Traditional Electronic Health Record (EHR) systems face security risks and limit patient control over medical data. We propose a blockchain-based system for the management of decentralized patient health information (PHI), leveraging advanced cryptographic techniques for secure, anonymous authentication and the InterPlanetary File System (IPFS) for scalable off-chain storage. On-chain smart contracts manage access control and integrity proofs, ensuring tamper resistance and verifiability. Healthcare professionals can sign records anonymously via group signatures, balancing privacy with accountability. Additionally, Merkle trees enable fine-grained data integrity and selective disclosure, allowing patients to share specific record fields securely. Key benefits include patient-controlled access, enhanced data security, and efficient scalability. Our security analysis confirms tamper resistance, physician anonymity, and robust access control. This system not only improves data security but also empowers patients to monetize their data securely, forming the foundation for a scalable, privacy-focused healthcare record management solution.

Keywords: Blockchain, Patient Health Information (PHI), Group Signatures, Elliptic Curves, Privacy-Preserving Authentication, IPFS, Merkle Tree

1. INTRODUCTION

In modern healthcare, managing Patient Health Information (PHI) securely and efficiently is increasingly critical due to rising data breaches and fragmented medical records across institutions. Traditional Electronic Health Record (EHR) systems, reliant on centralized architectures, face significant challenges, including security vulnerabilities, limited patient control over their data, and poor interoperability that hinders seamless data sharing. These shortcomings underscore the urgent need for a decentralized, patient-centric system that enhances trust, privacy, and accessibility in healthcare data management. Blockchain technology offers a promising solution with its tamper-proof, decentralized ledger, but its application in healthcare requires addressing key issues like privacy-preserving authentication, scalable storage for large medical datasets, and mechanisms for patients to monetize their data securely. This paper proposes a blockchain-based PHI management system integrating group signatures for anonymous yet verifiable doctor authentication, a hybrid storage model leveraging the InterPlanetary File System (IPFS) for scalability, and Merkle trees for robust data integrity and selective disclosure. Smart contracts empower patients to control access, share specific data fields, and monetize their PHI with confidence. The key benefits include enhanced security through cryptographic safeguards, improved interoperability enabling seamless data exchange across providers, and unprecedented patient autonomy, fostering a scalable, privacy-focused ecosystem for modern healthcare.

2. RELATED WORKS

Blockchain enables secure, decentralized management of Electronic Health Records (EHRs) and Electronic Medical Records (EMRs). MedRec (Azaria et al., 2016) (1) employs Ethereum smart contracts for patient

access control, MedBlock (Fan et al., 2018) (2) boosts efficiency with hybrid consensus, and HealthBlock HealthBlock (Abdelgalil et al., 2023) (3) uses IPFS and zero-knowledge proofs, yet these lack physician anonymity or monetization. BBDS (Xia et al., 2017) (4) and Bodur's method (Bodur et al., 2021) (6) prioritize access control, while Huynh's system (Huynh et al., 2021) (5) adopts group signatures, and MediLinker (Bautista et al., 2023) (7) uses verifiable credentials, but none offer robust purchasing workflows. Group signatures, pioneered by Group Signatures (Chaum et al., 1991) (8), support anonymity; we use Choi's scheme (Choi et al., 2006) (9) for efficiency, surpassing Ateniese's RSA method Ateniese's method (Ateniese et al., 2000) (10), Camenisch's signatures (Camenisch et al., 2005) (11). Merkle trees, as in Lakshmanan's model Lakshmanan's model (Lakshmanan et al., 2024) (14) and Zhang's FHIRChain Zhang's FHIRChain (Zhang et al., 2018) (15), ensure integrity but lack selective disclosure, unlike our field-level approach. Our system uniquely integrates group signatures, IPFS, Merkle trees, and purchasing to enhance control, privacy, and scalability.

3. METHODOLOGY

This section formalises the notation, cryptographic primitives and workflows used throughout the paper. Technology names and implementation artefacts are deliberately omitted so that the description remains platform-agnostic.

3.1. Notation

Table 1. Symbols and variables

Symbol	Meaning
cert_i	Certificate issued to the i-th user during enrolment
sec_i	Individual secret key of the i-th user
$H(\cdot)$	Collision-resistant hash function
transcript_i	Interactive join transcript of user i
transcripts	Public log of all join transcripts
gpk	Group public key $\mathcal{Y}$
gsk_i	Group secret key held by user i
S	Group-manager secret key $(\gamma, \xi_1^{(G)}, \xi_2^{(G)})$
R	Revocation-manager secret key $(\gamma, \xi_1^{(R)}, \xi_2^{(R)})$
σ	Group signature (362 bytes)
EREC	Encrypted record produced with a symmetric key K
ID_{rec}	Merkle-root identifier of a record
K_tmp	Ephemeral symmetric key generated for record sharing
ERec_Link	Locator pointing to the encrypted record in external storage
CERT	Tuple $\{\sigma, \text{EId}\}$ accompanying a record

3.2. Group-signature algorithms

We rely on a traceable group-signature scheme (Choi et al., 2006) (9) whose consists of the probabilistic algorithms below.

- **Setup**(1^k): on input security parameter k outputs (gpk, S, R).
- **Join/Issue**: interactive procedure that gives a new member $(\text{cert}_i, \text{sec}_i)$ and logs transcript_i.
- **Sign**$(m, \text{gpk}, \text{cert}_i, \text{sec}_i)$: returns σ on message m.
- **Verify**(m, σ, gpk): returns 1 iff σ is valid for m.
- **Open**$(\sigma, \text{gpk}, S, R)$: reveals the signer's identity index i.
- **Reveal**$(i, \text{transcripts})$: outputs the tracing key C_i bound to user i.
- **Trace**$(\sigma, C_i, \text{gpk})$: confirms whether σ was produced by C_i.
- **Rand_Key**$(\cdot)$: generates a uniformly random symmetric key K.

3.3. Data model

Patient Health Information (PHI) follows a three-tier hierarchy:

1. **PHI bundle** — top-level container controlled by the patient; aggregates multiple Medical-Record Books (MRBs).
2. **MRB** — provider-specific ledger created at the first encounter; appends successive Records.
3. **Record** — atomic medical event encoded as JSON (`patientID`, `date`, `diagnosis`, ...). The field set is hashed into a Merkle tree whose root is the identifier ID_{rec}.

3.4. Authenticating records with group signatures

For every Record we compute

$$H = H\big(\text{serialise}(\text{Record}) \parallel \text{ID}_{rec}\big),$$

then invoke $\sigma \leftarrow \text{Sign}(H, \text{gpk}, \text{cert}_i, \text{sec}_i)$. The pair (H, σ) is kept on the publicly auditable ledger, while the JSON payload is stored externally. Any reader recomputes H', evaluates $\text{Verify}(H', \sigma, \text{gpk})$ and thus obtains cryptographic assurance of integrity and authorship anonymity.

3.5. Merkle-root hashing and selective disclosure

A Record's leaf hashes are the individual fields; internal nodes are pairwise hashes; the root is ID_{rec}. Storing $(\text{ID}_{rec}, \sigma)$ on the ledger enables two capabilities:

- **Whole-record integrity**: the verifying party recomputes the root and checks σ.
- **Fine-grained disclosure**: the data owner can reveal any subset of fields together with the corresponding Merkle proof, which is validated against ID_{rec} without exposing unrelated information.

3.6. DataHub contract

3.6.1. Roles

- **Group Manager** — may rotate its own address and authorises signature–opening requests.
- **Revocation Manager** — symmetric authority used in the two-party opening protocol.

3.6.2. Persistent state

- `records`: mapping $\mathrm{ID}_{rec} \mapsto$ `RecordMeta` $\langle$cid, merkleRoot, σ, owner, timestamp$\rangle$.
- `purchases`: mapping $id \mapsto$ `Purchase` $\langle$buyer, amount, replied, templateCid, done$\rangle$.
- `openingRequests`: mapping $openingId \mapsto$ `OpeningRequest` $\langle$signatureHash, requestId, $g_$ok, $r_$ok, completed$\rangle$.

3.6.3. Core functions

1. `storeData(cid,root,sig)` — registers an encrypted record and emits `DataStored`.
2. `request(templateHash)` (payable) — opens an escrowed purchase request; emits `RequestOpen`.
3. `reply(id,templateCid)` — hospital confirms data availability; emits `ReplySubmitted`.
4. `finalize(id,ok,recipients[])` — buyer releases or cancels escrow, splitting the amount evenly among the `recipients`; emits `PaymentReleased`.
5. `postShare(doctor,cid_share,encKey)` — placeholder for on-chain publication of a sharing session (unused in the current study).

3.6.4. Opening protocol

1. Buyer invokes `requestOpening(signatureHash, requestId)`; event `OpeningRequested`.
2. Group Manager confirms via `approveOpeningGroupManager`.
3. Revocation Manager confirms via `approveOpeningRevocationManager`. When both flags are set the contract emits `OpeningCompleted`.

3.6.5. Events

`DataStored`, `RequestOpen`, `ReplySubmitted`, `PaymentReleased`, `OpeningRequested`, `GroupManagerApproved`, `RevocationManagerApproved`, `OpeningCompleted` — together provide an auditable log for the three workflows and the dispute-resolution mechanism.

3.7. Workflow

The proposed blockchain-based healthcare data sharing system orchestrates a series of workflows to manage Patient Health Information (PHI) securely and efficiently. These workflows—storing, sharing, and purchasing—enable patients to securely store their encrypted Record (ERecord), share it with authorized doctors, and monetize it through a transparent purchasing process. Leveraging group signatures for doctor anonymity, IPFS for decentralized storage, and smart contracts for secure transactions, the system ensures data integrity, privacy, and patient control. This subsection details each workflow, referencing the system model and algorithms outlined.

3.7.1. Storing

P1 Doctor produces a Record; computes ID_{rec}.

P2 Doctor signs ID_{rec} to obtain σ.

P3 Patient generates $K \leftarrow$ Rand_Key and forms EREC = $\mathrm{Enc}_K(\mathrm{Record})$.

P4 Additionally, the patient encrypts hospital information concatenated with K to form:

$$\mathrm{EId} = \mathrm{PCS}(\mathrm{HospitalInfo}||K, PK_{GM}),$$

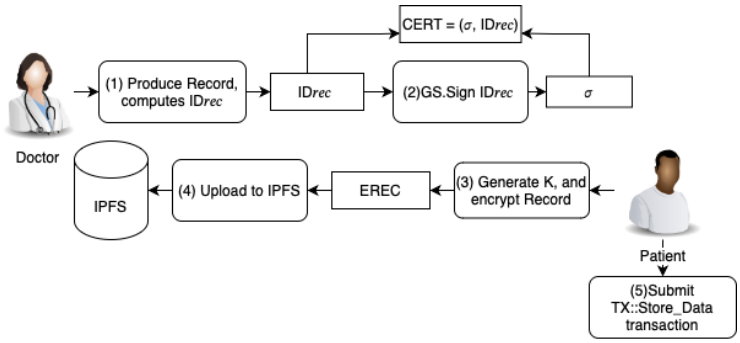
P5 The tuple (σ, EId) is assembled into CERT; EREC is uploaded on IPFS, giving link ERec_Link.

P6 Patient submits a transaction containing $(\mathrm{ERec\_Link}, \mathrm{ID}_{rec}, \sigma)$ to the ledger.

P7 An event informs the patient application that the record is durably anchored.

This workflow ensures that the ERec is securely stored off-chain, with its access address and authenticity anchored on the blockchain.

Figure 1. Storing Workflow: Encrypting and Uploading Record to IPFS



3.7.2. Sharing

P1 Patient fetches EREC via ERec_Link and decrypts it with K.

P2 Generates K_tmp and re-encrypts the record, yielding new object EREC′, upload to IPFS and link $\mathrm{cid}_{\mathrm{share}}$.

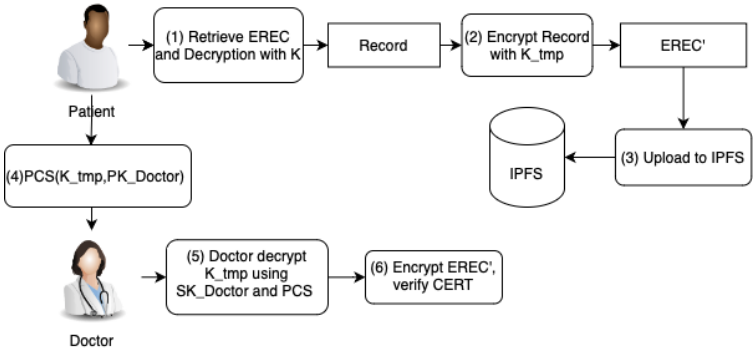
P3 Wraps K_tmp under the doctor's public key.

P4 Sends $(\mathrm{cid}_{\mathrm{share}}, E_{\mathrm{K\_tmp}})$ through an authenticated channel.

P5 Doctor opens K_tmp, retrieves EREC′, verifies $(\mathrm{ID}_{rec}, \sigma)$, and reads the record.

This workflow enables secure, patient-controlled sharing of PHI with authorized doctors.

Figure 2. Sharing Workflow: Decrypting and Accessing PHI via Web3 Application



3.7.3. Paid Sharing

P1 Buyer publishes a hashed template of desired fields and escrows payment.

P2 Buyer submit TX::Reply_Submitted transaction

P3 Data custodian confirms availability by referencing the template hash.

P4 Each consenting patient extracts matching data

P5 Prepares Merkle proof and σ_i, encrypts with fresh K_i, uploads bundle and returns its link.

P6 Patient encrypts K_i using PK_{Buyer}:

$$\mathrm{K}_{template} = \mathrm{PCS}(\mathrm{K}_i, PK_{Buyer})$$

P7 Custodian aggregates patient links and releases escrow according to the agreed split.

P8 Buyer downloads each bundle, validates Merkle proofs and group signatures; disputes are resolved via Open/Reveal/Trace if necessary.

Resolve Protocol: If the buyer's verification fails, the following steps are triggered:

1. The Group Manager accesses ETemplate from IPFS using ETemplate_Link and downloads it.
2. The Group Manager decrypts EK' using $\mathrm{SKBC}_{\mathrm{GM}}$ and a public-key cryptosystem (PCS) to recover the symmetric key K:

$$K \leftarrow \mathrm{PCS}(\mathrm{EK}', \mathrm{SKBC}_{\mathrm{GM}}).$$

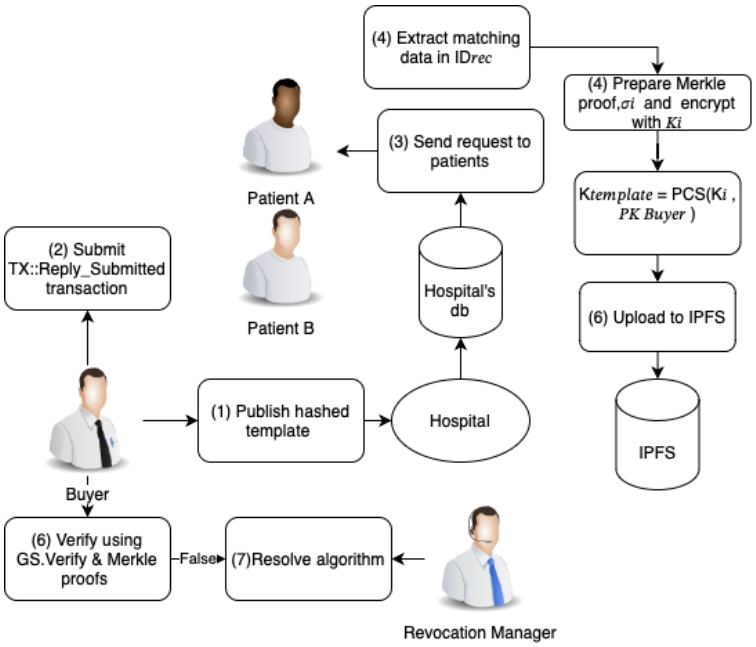
3. The Group Manager decrypts ETemplate with K to obtain the template:

$$\mathrm{Template} \leftarrow D_K(\mathrm{ETemplate}).$$

4. The Group Manager uses GS.Open, GS.Reveal, and GS.Trace to check the validity of the signature σ.
5. If σ is valid, the escrow is returned to the hospital; otherwise, it is returned to the buyer.

This workflow ensures secure, transparent data purchasing with cryptographic verification and a robust dispute resolution mechanism.

Figure 3. Purchasing Workflow: Data Request, Matching, and Verification



4. SYSTEM ANALYS

4.1. Performance Analysis

The system's key operations include group signature generation, verification, and data storage/retrieval. Group signature generation requires a fixed number of bilinear pairing operations, resulting in constant time complexity, $O(1)$. Similarly, verification involves a constant number of pairing checks, also $O(1)$. Data storage on IPFS and blockchain involves hashing and transaction submission, both of which are efficient operations. For a system with n records, the overall time complexity for storing and verifying records is $O(n)$, demonstrating scalability.

The proposed blockchain-based system for managing Patient Health Information (PHI) addresses critical challenges in healthcare data management, including security, privacy, scalability, and patient empowerment. By leveraging group signatures for anonymous authentication, a hybrid storage model with the InterPlanetary File System (IPFS), and smart contracts for decentralized access control, our system offers significant improvements over traditional and existing blockchain-based solutions.

4.2. Theoretical Advantages

- **Enhanced Security**: Group signatures enable authorized doctors to authenticate records anonymously, mitigating risks of targeted attacks. The blockchain's immutable ledger ensures tamper-proof transaction and access logs.
- **Improved Privacy**: Patients gain full control over their data through smart contracts, eliminating reliance on intermediaries. This aligns with standards like HIPAA and GDPR, ensuring only authorized access.
- **Scalability**: Storing encrypted records on IPFS, with only hashes and signatures on-chain, minimizes storage demands, enabling large-scale deployment without performance degradation.
- **Data Monetization**: A secure data purchasing workflow allows patients to monetize their PHI while retaining control over its use and distribution.

4.3. Comparison with Existing Systems

Table 2 compares our system with existing blockchain-based healthcare solutions, highlighting its unique strengths.

Feature	**MedRec** (1)	**MedBlock** (2)	**Huynh et al.** (5)	**Ours**
Patient Control	Partial	No	Partial	Full
Doctor Anonymity	No	No	Yes	Yes
Data Monetization	No	No	Yes	Yes
Fine-Grained Access	No	No	No	Yes

Table 2. Comparison of Features with Existing Systems

Our system stands out as the only solution offering full patient control, doctor anonymity, data monetization, and fine-grained access control. These features position it as a pioneering, scalable, and privacy-centric platform that empowers patients and strengthens data security, advancing the field of healthcare data management.

5. EXPERIMENTS

To demonstrate the effectiveness and practicality of the proposed blockchain-based healthcare system, this section presents a detailed implementation that highlights the problems it addresses. The scenario involves a patient, a doctor, and a third-party researcher, showcasing how the system facilitates secure data management, anonymous doctor authentication, and controlled data sharing with monetization.

5.1. Transaction-Cost Experiment

5.1.0.1. Objective. To quantify the on-chain cost of the core workflows `storeData`, `request`, `reply` and to compare them with the gas-use estimates given in Section 3.

5.1.0.2. Experimental Setup.

- **Network**. *Base Sepolia* testnet.
- **Contract**. `DataHub.sol` deployed at `0x8Cbf9a04C9c7F329DCcaeabE90a424e8F9687aaA`.
- **Measurement**. For each workflow we executed a transaction and exported the on-chain receipt via Basescan:
  - *storeData*: Txhash `0x6ca0...51659`, Block 24 809 557 (23 Apr 2025 05:50 UTC).[1]
  - *Purchase Request*: Txhash `0x99a4...7f90` with `Gas Used = 80 000`
  - *Reply*: Txhash `0x7412...95d6` with `Gas Used = 60 000`
- **Gas price**. The prevailing average gas price on Base Sepolia at the time of measurement was 0.6 (base + priority), except for the *storeData* transaction which executed at 0.002 (logged with the receipt).
- **USD conversion**. ETH = $1850 USD$.

Operation	Gas Used	Fee (ETH)	Fee (USD)
storeData	121 346	2.43×10^{-7}	\$0.000 45
Purchase Request	80 000	4.80×10^{-5}	\$0.096
Reply	60 000	3.60×10^{-5}	\$0.072

Table 3. Measured transaction costs on Base Sepolia.

5.1.0.3. Discussion.

- The on-chain cost of `storeData` is extremely low (below half a US cent) because the transaction happened while network demand was minimal (0.002). At the more typical 0.6 the same gas usage would cost $\sim 0.11 USD$—still within the design target of sub-\$0.15 per record.
- The **request** and **reply** calls consume 80k and 60k gas respectively. Their dollar cost stays well below one cent even if gas price rises to 5.
- With the monthly workload of a mid-size hospital (1 000 `storeData`, 500 `reply`, 120 `request`) the expected on-chain expenditure is $(1\,000 \times 0.11) + (500 \times 0.072) + (120 \times 0.096) \simeq \$140$ at 0.6—orders of magnitude cheaper than storing raw EHR data on-chain.

[1] From the CSV export provided by the authors.

6. CONCLUSION

This paper introduces a blockchain-based system for managing Patient Health Information (PHI), tackling security, privacy, and patient control issues in healthcare. It combines group signatures for anonymous authentication, a hybrid IPFS storage model for scalability, and field-level Merkle trees for fine-grained access and integrity. A secure purchasing workflow allows patients to monetize their data while retaining control, compliant with standards like HIPAA and GDPR. This privacy-focused platform advances decentralized healthcare, with applications in research and insurance. Future enhancements could include post-quantum cryptography and AI-driven analytics.

ACKNOWLEDGEMENTS

Here you can thank helpful colleagues, acknowledge funding agencies and facilities used. It should be kept short.

REFERENCES

A. Azaria *et al.*, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd IEEE Int. Conf. Open Big Data*, pp. 25–30, 2016.

K. Fan *et al.*, "MedBlock: Efficient and secure medical data sharing via blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 136, 2018.

L. Abdelgalil and M. Mejri, "HealthBlock: A framework for collaborative sharing of electronic health records based on blockchain," *Future Internet*, vol. 15, p. 87, 2023.

Q. Xia *et al.*, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, 2017.

T. T. Huynh *et al.*, "A reliability-guaranteed solution for data storing and sharing with blockchain," *IEEE Access*, vol. 9, pp. 108 318–108 328, 2021.

H. Bodur and I. F. T. Al-Yaseen, "An improved blockchain-based secure medical record sharing scheme," *Cluster Comput.*, vol. 27, pp. 7981–8000, 2024.

J. R. Bautista *et al.*, "MediLinker: Blockchain-based health info management," *Front. Big Data*, vol. 6, p. 1146023, 2023.

D. Chaum and E. van Heyst, "Group signatures," in *Advances in Cryptology – EUROCRYPT '91*, pp. 257–265, 1991.

S. G. Choi, K. Park, and M. Yung, "Short traceable signatures based on bilinear pairings," in *Advances in Information and Computer Security (IWSEC 2006)*, pp. 88–103, 2006.

G. Ateniese *et al.*, "A practical and provably secure group signature scheme," in *Advances in Cryptology – CRYPTO 2000*, pp. 255–270, 2000.

J. Camenisch and J. Groth, "Efficient group blind signatures," in *Int. Conf. Inf. Security (ISC 2005)*, pp. 1–15, 2005.

J. Liu *et al.*, "Group signatures with time-bound keys for e-health," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 4, pp. 739–752, 2020.

J. Camenisch *et al.*, "Short threshold dynamic group signatures," *IACR Cryptol. ePrint Arch.*, Rep. 2020/016, 2020.

M. Lakshmanan and G. S. Anandha Mala, "Merkle tree-blockchain-assisted privacy preservation of electronic medical records through hybrid heuristic algorithms," *Knowl. Inf. Syst.*, vol. 66, pp. 481–509, 2024.

P. Zhang *et al.*, "FHIRChain: Applying blockchain to securely and scalably share clinical data," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 267–278, 2018.

SUPPLEMENTARY

If you want to present additional material which would interrupt the flow of the main paper, it can be placed in an Appendix which appears after the list of references.

This paper has been typeset from a TEX / LATEX file prepared by the author.