



[A Blockchain-Based Framework for Securing Cloud-Based Electronic Health Records Using Smart Contract Access Control, IPFS and Elliptic Curve Cryptography]

Pramod Kumar Yadav*1, Dr. Vivek Kumar*2

*1 Research Scholar, Mahakaushal University, Jabalpur, (M.P.)

*2 Professor, Mahakaushal University, Jabalpur, (M.P.)

ABSTRACT

Cloud computing enables healthcare organizations to store and share electronic health records (EHRs) at low cost, but centralized cloud storage exposes sensitive patient data to unauthorized access, tampering, single points of failure and dependence on trusted third parties. This paper proposes a novel approach for securing cloud-based EHR systems using blockchain technology. The framework combines a smart-contract-based access control model managed by an EHR manager and a policy store, a decentralized storage layer based on the InterPlanetary File System (IPFS) with a distributed hash table (DHT), and elliptic curve cryptography (ECC) for encrypting patient records with short keys suitable for resource-constrained devices. The data uploading and data sharing phases are specified, together with a use-case model involving healthcare providers, cloud service providers and end users. Network performance is modelled in terms of file loss and total transmission time, and security is analysed through the probability that an attacker obtains all replicated file blocks and the probability that an attacker controlling part of the network rewrites the chain. The analysis shows that the attack success probability falls below 0.1% after five confirmation blocks when the attacker controls 10% of the computing power. The prototype, implemented on Ethereum with Solidity smart contracts, Remix IDE, web3js, Infura and JSON-RPC, and audited with SmartCheck and Oyente, supports registration, key generation, encrypted upload, block creation, user verification, secret key verification and decryption. A comparison with four existing schemes shows that the proposed system is the only one that satisfies all seven criteria of confidentiality, availability, decentralized access, authorization, user authentication, integrity and privacy, while achieving lower network latency for 100 to 500 users.

KEYWORDS: blockchain; electronic health records; cloud computing; elliptic curve cryptography; IPFS; smart contracts; access control; data security

Received

25/07/2026

Accepted

15/09/2026

Published

19/09/2026

Article Type

Research Paper



1. INTRODUCTION

The transition from paper-based to electronic medical records has transformed healthcare delivery. Electronic health records (EHRs) aggregate patient demographics, clinical notes, laboratory and imaging reports, prescriptions, billing and insurance information, and increasingly data from wearable and IoT devices, enabling better clinical decisions, fewer medical errors and lower costs. Telecare medicine information systems further allow clinicians to deliver remote care on the basis of these records. Cloud computing, with its on-demand, elastic and shared resources [1], is an attractive platform for storing EHRs and sharing them among hospitals, laboratories, insurers and researchers.

However, outsourcing EHRs to the cloud raises serious security and privacy concerns. Records held by a cloud service provider can be accessed or altered without the knowledge of the patient, the provider itself may be a target of attack or a single point of failure, and users must trust third parties for authentication and auditing. The security of EHR data therefore requires confidentiality, integrity and availability, together with authentication, fine-grained authorization and accountability. Trust, in particular, has been identified as one of the most significant barriers to sharing electronic health data in the cloud.

Blockchain technology offers a promising solution. A blockchain is a tamper-evident ledger replicated across many nodes, in which each block is linked by a cryptographic hash to its predecessor, and consensus prevents any single party from altering records [2]. Smart contracts can encode access control policies that execute automatically [3]. These properties have motivated a growing body of work on blockchain-based medical data management [4]–[7], beginning with MedRec [8] and followed by systems for cloud-based data sharing [9], privacy-preserving access control [10], efficient medical data sharing [11] and secure storage [12].

Existing schemes, however, often address only part of the security requirements, rely on heavyweight cryptography, or store large medical files directly on-chain. This paper proposes a novel approach that secures cloud-based EHR systems with blockchain by combining smart-contract access control, decentralized IPFS storage and ECC encryption. The main contributions are: (i) a system architecture with an EHR manager, policy-based smart contracts, an administrator and decentralized storage, together with detailed data uploading and data sharing protocols; (ii) a use-case model defining the responsibilities of healthcare providers, cloud service providers and end users; (iii) a network performance and security analysis based on file loss, transmission time and attack probabilities; and (iv) an Ethereum-based implementation and a comparative evaluation of security features and network latency against existing schemes.

2. RELATED WORK AND BACKGROUND

2.1 Blockchain-Based EHR Management

Azaria *et al.* [8] proposed MedRec, which uses Ethereum smart contracts to record pointers to medical data and manage permissions across providers. Xia *et al.* [9] presented MeDShare, which enables trust-less data sharing among cloud service providers and logs data access on a blockchain. Dagher *et al.* [10] developed Ancile, which uses smart contracts and proxy re-encryption to provide access control and interoperability. Fan *et al.* [11] proposed MedBlock, which improves consensus efficiency and combines access control with symmetric encryption, while Chen *et al.* [12] used blockchain and cloud storage together to secure medical records and support medical services. Nguyen *et al.* [13] addressed EHR sharing in mobile cloud e-health systems using smart contracts and IPFS, and Dwivedi *et al.* [14] designed a privacy-preserving healthcare blockchain tailored to IoT devices. More recently, Zaabar *et al.* [15] presented HealthBlock, a secure healthcare data management system based on blockchain and off-chain storage.

Several schemes are closely related to the present work and are used for comparison. Rifi *et al.* [16] examined blockchain for eHealth data access management and proposed off-chain storage of large medical data. BaDS [17] combined blockchain with attribute-based signatures and ciphertext-policy attribute-based encryption for user-controlled data sharing in IoT. Table 1 summarizes the characteristics of representative schemes.



Table 1. Summary of representative blockchain-based medical data management schemes

Scheme	Platform / storage	Access control	Cryptography	Main limitation
MedRec [8]	Ethereum; off-chain provider databases	Smart contracts	Hash pointers	No encryption of stored records
MeDShare [9]	Cloud with blockchain audit log	Smart contracts	Signatures	Relies on cloud provider trust
Ancile [10]	Ethereum; off-chain	Smart contracts	Proxy re-encryption	High computational cost
MedBlock [11]	Private blockchain	Access control lists	Symmetric encryption	Limited decentralization
BaDS [17]	Blockchain with cloud	Attribute policies	ABS, CP-ABE	Expensive attribute-based operations
Proposed	Ethereum; IPFS with DHT	Smart contracts, policy store	ECC	—

2.2 Blockchain Structure

A blockchain is a sequence of blocks, each containing a timestamp, a block number, the block hash, the block data and the hash of the previous block (Fig. 1). Because the hash of each block depends on the contents of the previous block, any modification of historical data changes all subsequent hashes and is immediately detectable by other nodes [2]. New blocks are added only after validation by miners or validators according to the consensus protocol, so that all nodes maintain an identical copy of the ledger.

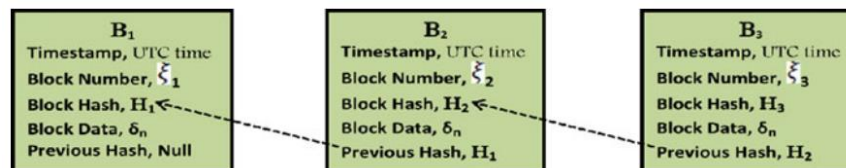


Fig. 1. Structure of a blockchain: each block stores a timestamp, block number, block hash, data and the hash of the previous block.

2.3 Elliptic Curve Cryptography

Elliptic curve cryptography was proposed independently by Miller [18] and Koblitz [19]. It is a public-key technique based on the algebraic structure of points on an elliptic curve $y^2 = x^3 + ax + b$ over a finite field. Given two points P and Q on the curve, the line through them intersects the curve at a third point, whose reflection defines $R = P + Q$ (Fig. 2). Repeated addition defines scalar multiplication kP , and the security of ECC rests on the elliptic curve discrete logarithm problem: given P and kP , it is computationally infeasible to recover k . Because no sub-exponential algorithm is known for this problem, ECC achieves the same security as RSA with much shorter keys (Table 2) {nist57}, which reduces computation, memory, bandwidth and energy consumption and makes ECC particularly suitable for mobile and IoT devices used in healthcare.

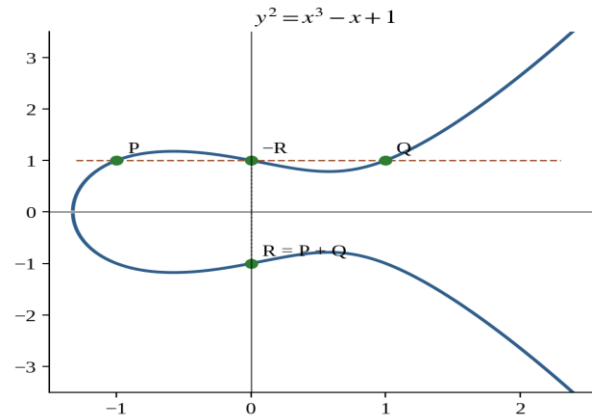


Fig. 2. Geometric illustration of point addition on the elliptic curve $y^2 = x^3 - x + 1$, the basis of ECC scalar multiplication.

Table 2. Comparable security strengths of symmetric, RSA and ECC keys (NIST SP 800-57)

Security strength (bits)	Symmetric key (bits)	RSA modulus (bits)	ECC key (bits)	RSA : ECC key-size ratio
112	112 (3DES)	2048	224–255	$\approx 9 : 1$
128	128 (AES-128)	3072	256–383	$\approx 12 : 1$
192	192 (AES-192)	7680	384–511	$\approx 20 : 1$
256	256 (AES-256)	15360	512+	$\approx 30 : 1$

3. PROPOSED SYSTEM

3.1 System Architecture

The proposed cloud-based EHR system is shown in Fig. 3. Each EHR contains a patient's private health information and is identified by a patient ID and an area ID. The system consists of the following components.

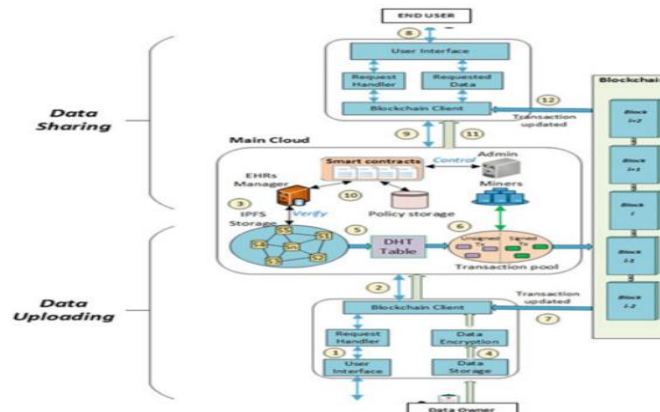


Fig. 3. Architecture of the proposed blockchain-based cloud EHR system showing the data uploading and data sharing phases.



The EHR manager mediates the secure exchange of data between legitimate users and the cloud server. It monitors user transactions, controls data storage procedures, grants data access and helps enforce strict user policies through smart contracts. Smart contracts define the operations required for access control; users interact with them through the application binary interface and the contract address, and the contracts authenticate requests and assign appropriate rights. The administrator adds and removes users, changes access permissions, revokes rights and manages transactions. Decentralized storage is provided by IPFS [20], because the blockchain itself cannot store large medical files efficiently. IPFS offers high storage throughput, faster retrieval and no single point of failure, and a DHT maintains the hash values of all medical records so that encrypted files can be located by their content hashes.

3.2 Data Uploading Phase

When a data owner (healthcare provider) wishes to upload records, it first registers and sends a request to the cloud server, which invokes the EHR manager. The EHR manager forwards the request to the smart contract, which validates it against the policy store to confirm that the uploader is authorized, and returns an acknowledgement. The data file is then encrypted with ECC using the public key and uploaded to the IPFS server. IPFS automatically computes the content hash of the file, returns it to the EHR manager and records it in the DHT. The metadata of the submitted file are placed in the transaction pool, from which miners create and verify blocks and propagate them to other nodes; once consensus is reached, the block is appended to the chain and every node can independently verify the transaction.

3.3 Data Sharing Phase

An end user must register on the blockchain and receives a public-private key pair. The user's request is authenticated by the cloud server and passed to the EHR manager, which invokes the smart contract to validate the request against the policy list. If the user is authorized, the EHR manager looks up the hash value in the DHT using the area ID and patient ID, retrieves the encrypted record from IPFS and returns it; the record can be decrypted only with the user's private key. Requests from unauthorized users are rejected by the EHR manager. As in the uploading phase, each sharing transaction is added to the transaction pool, mined into a block and confirmed by other miners, creating an auditable record of every data access. The complete message flow is shown in Fig. 4 and formalized in Algorithm 1.

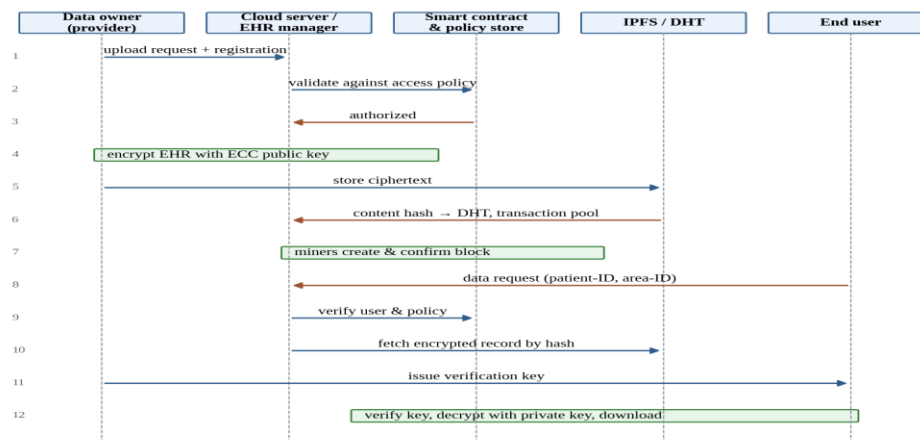


Fig. 4. Message sequence of the data uploading (steps 1–7) and data sharing (steps 8–12) phases.



Algorithm 1. Protocol for accessing EHR data

Input: request from user U for record of patient-ID p in area-ID a

Output: decrypted EHR or access denial

U registers on the blockchain and obtains (PK_U, SK_U)

cloud server authenticates U and forwards request to EHR manager

EHR manager invokes smart contract: check policy store for (U, p, a)

if policy check succeeds then

$h \leftarrow \text{DHT.lookup}(p, a)$

$C \leftarrow \text{IPFS.get}(h)$

 data owner issues verification key to U

 if verification key is valid then

$M \leftarrow \text{ECC.decrypt}(C, \text{SK}_U)$; return M to U

 record access transaction in pool; miners append block

else

 deny request and log failed attempt

3.4 Use-Case Model

The system involves three types of actors (Fig. 5). The healthcare provider (data owner) must register with the cloud service provider (CSP) with accurate information, receives a cloud ID, generates keys, creates a block for each submission and uploads encrypted files. The CSP verifies the provider's information, authenticates and validates each matching request and supplies verification keys. The end user views and requests files, receives authentication and a verification key after the owner verifies the request, performs proof verification and downloads and decrypts the file.

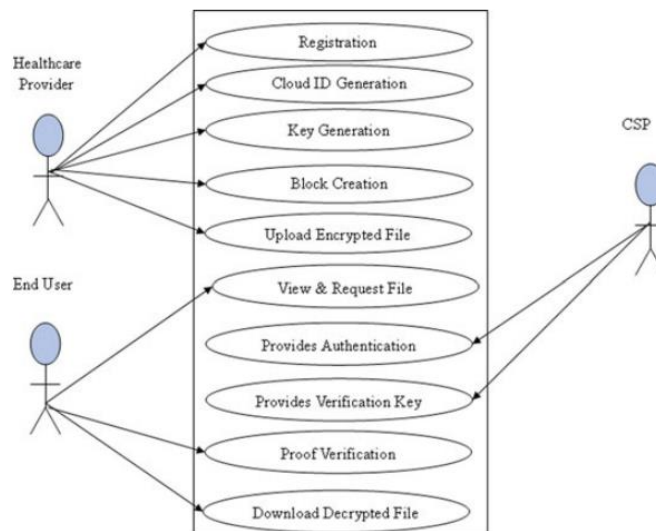


Fig. 5. Use-case diagram showing the roles of the healthcare provider, cloud service provider (CSP) and end user.



4. PERFORMANCE AND SECURITY ANALYSIS

4.1 Network Performance Model

Network performance is analysed in terms of file loss and transmission time; the objective is to reduce latency and file loss while enhancing security. The notations used are listed in Table 3.

Table 3. Notations used in the performance and security analysis

Notation	Meaning	Notation	Meaning
LFS	Lost file size	FLP	File loss probability
TTF	Total transmission time of file	FS	File size
b_s	Block size	b_w	Bandwidth between user and data centre
BWS_k	Latency of link to node k	N	Number of P2P network nodes
r	Number of malicious nodes	m	Number of file blocks
K	Probability that an honest node produces the next block	Q	Probability that the attacker produces the next block
z	Number of blocks the attacker must catch up	AT	Probability of a successful attack

Consider users connected to several blockchain data centres through paths of varying reliability. The expected lost file size is

$$L = LFS \cdot FLP \quad (1)$$

Each file block is 32 KB, and transmitting a block of size b_s over a link of bandwidth b_w incurs a delay of b_s/b_w . Selecting, for each block, the node N_k with the lowest transmission delay, the total transmission time of the file is

$$TTF = \sum_{k=1}^N \frac{FS}{\arg \min_k BWS_k} \quad (2)$$

4.2 Security Analysis

In the proposed design, replicas of each file block are stored randomly on nodes of the network. If r of the n nodes are malicious, the probability that an attacker obtains a replica of a given block is r/n , and the probability of obtaining all m blocks of a file is

$$P_{\text{replica}} = \left(\frac{r}{n}\right)^m \quad (3)$$

For large n and m , this probability becomes negligible. A stronger attacker may control a fraction of the network's computing power and attempt to create a longer alternative chain containing falsified records. Following the analysis of Nakamoto, the probability that the attacker ever catches up from z blocks behind is

$$Q_z = 1 \text{ if } K \leq Q; \quad (Q/K)^z \text{ if } K > Q \quad (4)$$



The attacker's progress while honest nodes produce z blocks follows a Poisson distribution with expected value $\lambda = z \cdot Q/K$. The overall probability of a successful attack is therefore

$$AT = 1 - \sum_{k=0}^z \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - \left(\frac{Q}{K}\right)^{z-k}\right), \quad \lambda = z \cdot \frac{Q}{K} \quad (5)$$

Combining Eqs. (3) and (5), the risk of data corruption in the system is

$$L = \left(\frac{r}{n}\right)^m \cdot AT \cdot \frac{b_s}{b_w} \quad (6)$$

Fig. 6 evaluates these expressions numerically. When the attacker controls 10% of the computing power, the attack success probability falls to 5.1% after two confirmations, 0.09% after five and about 10^{-6} after ten. Even with 30% of the computing power, it falls to 4.2% after ten confirmations and 0.25% after twenty. Similarly, when 30% of the nodes are malicious, the probability of obtaining all replicas of a ten-block file is below 10^{-5} . These results confirm that the risk of data corruption decreases exponentially with the number of blocks and file segments.

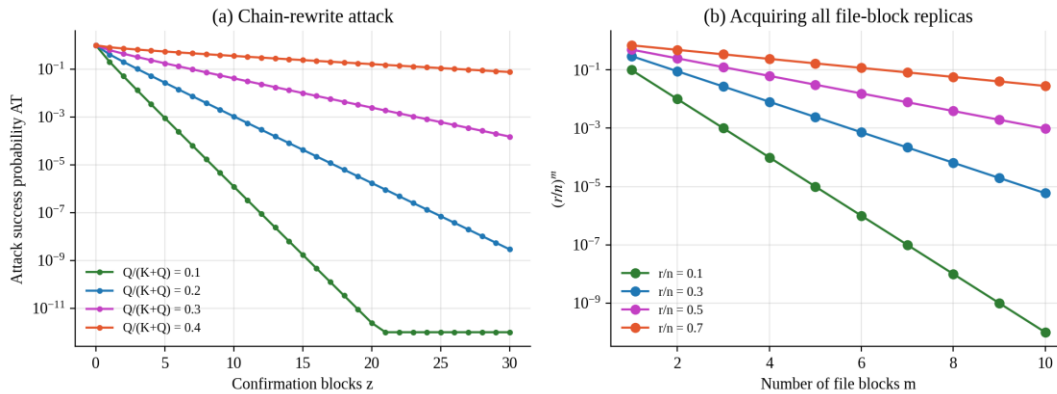


Fig. 6. Security analysis: (a) probability of a successful chain-rewrite attack versus the number of confirmation blocks for different attacker computing-power fractions and (b) probability of acquiring all replicas of an m-block file for different fractions of malicious nodes.

5. IMPLEMENTATION

The proposed approach was implemented on the public Ethereum blockchain. Smart contracts were written in Solidity and compiled and tested in the web-based Remix IDE, and the application communicated with the blockchain through web3js, Infura and JSON-RPC. Smart contract vulnerabilities were analysed with SmartCheck [21] at different severity levels, and contract security was further examined with Oyente [22]. The prototype implements the following workflow, illustrated in Fig. 7. A healthcare provider logs in, or registers if new, whereupon a cloud ID is generated automatically. The provider selects and uploads patient records (a dataset of 1000 records was used); generates an ECC key and encrypts the records; and sends a request to the cloud server. After the CSP approves the request, a blockchain is created with one block per record, and the encrypted file is uploaded to the cloud server, which issues an upload acknowledgement. The EHR manager views pending requests and verifies users before authorization. An end user searches for the required data and requests a download key from the healthcare provider, tracks the status of the request, and, after successful secret key verification, decrypts and downloads the file.

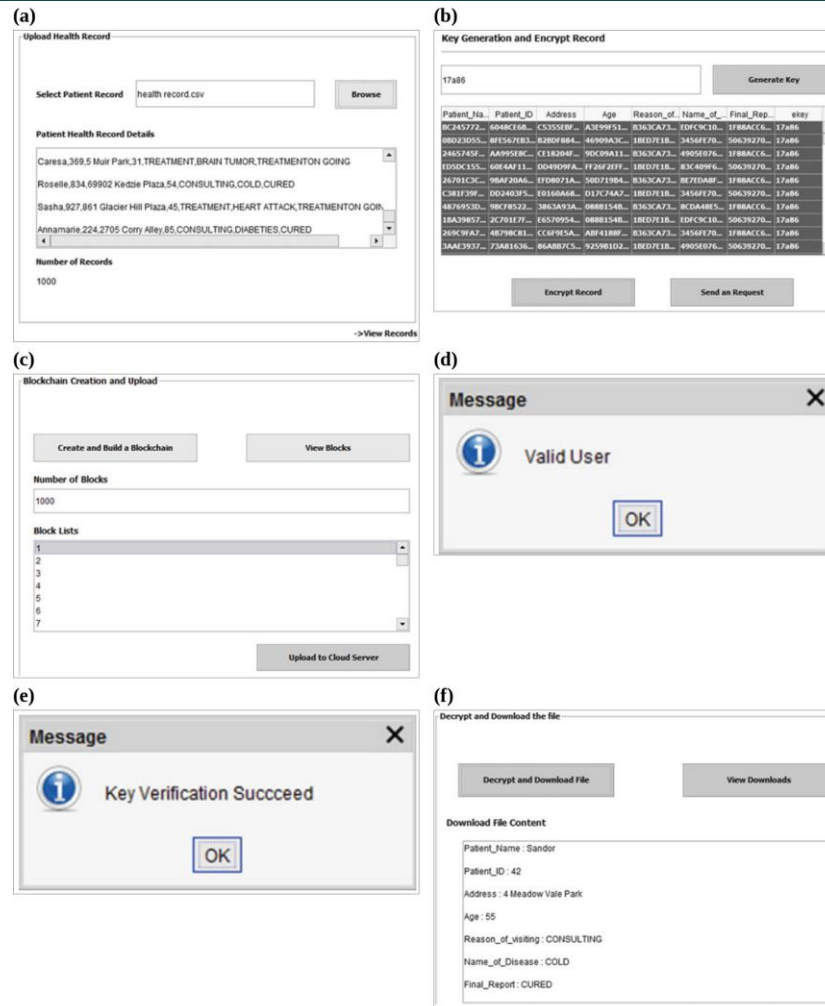


Fig. 7. Prototype interfaces: (a) selecting and uploading patient records, (b) key generation and record encryption, (c) blockchain creation and upload, (d) user verification, (e) secret key verification and (f) decryption and download of the file.

6. RESULTS AND DISCUSSION

6.1 Security and Privacy Features

The proposed system was compared with four existing schemes, namely those of Zhang *et al.*, BaDS [17], HealthTap and Rifi *et al.* [16], using seven security and privacy criteria (Table 4, Fig. 8). The proposed system satisfies all seven criteria. The schemes of Zhang *et al.* and BaDS do not provide data confidentiality or availability, BaDS additionally lacks data authorization, HealthTap lacks confidentiality, and the scheme of Rifi *et al.* lacks data authorization and user authentication. The combination of ECC encryption (confidentiality), IPFS replication (availability), smart contract policies (authorization and decentralized access), key-pair and verification-key checks (authentication) and hash-linked blocks (integrity) enables the proposed system to cover the full set of requirements.



Table 4. Comparison of security and privacy features with existing schemes

Feature	Zhang <i>et al.</i>	BaDS	HealthTap	Rifi <i>et al.</i>	Proposed
Data confidentiality	No	No	No	Yes	Yes
Availability	No	No	Yes	Yes	Yes
Decentralized access	Yes	Not reported	Yes	Yes	Yes
Data authorization	Yes	No	Yes	No	Yes
User authentication	Yes	Yes	Yes	No	Yes
Data integrity	Yes	Yes	Yes	Yes	Yes
Data privacy	Yes	Yes	Yes	Yes	Yes

Data confidentiality	X	X	X	✓	✓
Availability	X	X	✓	✓	✓
Decentralized access	✓	n/r	✓	✓	✓
Data authorization	✓	X	✓	X	✓
User authentication	✓	✓	✓	X	✓
Data integrity	✓	✓	✓	✓	✓
Data privacy	✓	✓	✓	✓	✓
	5/7	3/7	6/7	5/7	7/7
	Zhang <i>et al.</i>	BaDS	HealthTap	Rifi <i>et al.</i>	Proposed

Fig. 8. Security feature coverage of the proposed system and existing schemes (✓ provided, X not provided, n/r not reported); totals are shown below each column.

6.2 Network Latency

Network latency, the total time between a sender issuing a request and the recipient receiving and processing it, was measured for 100 to 500 users (Fig. 9). Latency increased approximately linearly with the number of users for all schemes. The proposed approach consistently showed the lowest latency, reaching approximately 2700 ms at 500 users, compared with roughly 2750–2800 ms for HealthTap and BaDS and about 3000–3200 ms for the schemes of Zhang *et al.* and Rifi *et al.* The advantage grows with load, which reflects the efficiency of ECC operations with short keys and the retrieval of content-addressed files from the nearest IPFS nodes rather than from a central server.

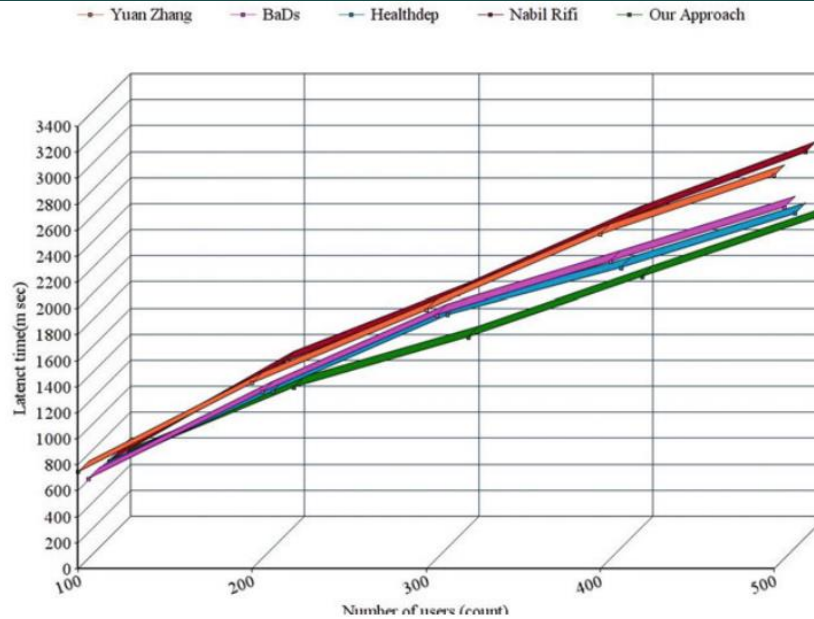


Fig. 9. Network latency of the proposed approach and existing schemes for different numbers of users.

6.3 Discussion

The results show that a design combining blockchain access control, decentralized storage and lightweight public-key encryption can meet the full set of EHR security requirements without sacrificing performance. Storing only hashes and metadata on-chain while keeping encrypted files in IPFS avoids the storage and cost limitations of on-chain data, in line with the approaches of Nguyen *et al.* [13] and HealthBlock [15]. Removing the need for the server to generate users' secret keys also reduces reliance on a trusted third party. Several limitations should be noted. The public Ethereum network incurs transaction fees and exposes transaction metadata, which may conflict with healthcare privacy regulations; a permissioned consortium blockchain may be preferable in practice [23]. The latency comparison is based on simulation with up to 500 users, and the security analysis assumes random placement of replicas and a Poisson model of attacker progress. Finally, key management and recovery for patients who lose their private keys requires further work.

7. CONCLUSION AND FUTURE WORK

This paper presented a novel blockchain-based approach for securing cloud-based EHR systems. The framework integrates an EHR manager and policy-based smart contracts for access control, IPFS with a DHT for decentralized encrypted storage, and ECC for efficient encryption with short keys. Data uploading and sharing protocols and a use-case model were specified, and analytical models showed that the probability of successful attacks decreases exponentially with the number of confirmation blocks and file blocks. The Ethereum prototype demonstrated the complete workflow from registration to decryption. Compared with existing schemes, the proposed system is the only one that satisfies all seven security and privacy criteria and achieves the lowest network latency for 100–500 users. Future work will deploy the framework on a permissioned blockchain, integrate attribute-based and patient-consent-driven access policies, add privacy-preserving analytics on EHR data, and evaluate the system with real hospital workloads and regulatory compliance requirements.

CONFLICT OF INTEREST

The author declares no conflict of interest.



REFERENCES

- [1] P. Mell and T. Grance, "The NIST definition of cloud computing," *NIST Special Publication 800-145*, 2011, doi: [10.6028/NIST.SP.800-145](https://doi.org/10.6028/NIST.SP.800-145).
- [2] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, pp. 557–564, 2017, doi: [10.1109/BigDataCongress.2017.85](https://doi.org/10.1109/BigDataCongress.2017.85).
- [3] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016, doi: [10.1109/ACCESS.2016.2566339](https://doi.org/10.1109/ACCESS.2016.2566339).
- [4] T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *J. Am. Med. Inform. Assoc.*, vol. 24, no. 6, pp. 1211–1220, 2017, doi: [10.1093/jamia/ocx068](https://doi.org/10.1093/jamia/ocx068).
- [5] M. Hölbl, M. Kompara, A. Kamišalić, and L. Nemec Zlatolas, "A systematic review of the use of blockchain in healthcare," *Symmetry*, vol. 10, no. 10, Art. no. 470, 2018, doi: [10.3390/sym10100470](https://doi.org/10.3390/sym10100470).
- [6] C. C. Agbo, Q. H. Mahmoud, and J. M. Eklund, "Blockchain technology in healthcare: A systematic review," *Healthcare*, vol. 7, no. 2, Art. no. 56, 2019, doi: [10.3390/healthcare7020056](https://doi.org/10.3390/healthcare7020056).
- [7] T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 135, pp. 62–75, 2019, doi: [10.1016/j.jnca.2019.02.027](https://doi.org/10.1016/j.jnca.2019.02.027).
- [8] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data (OBD)*, pp. 25–30, 2016, doi: [10.1109/OBD.2016.11](https://doi.org/10.1109/OBD.2016.11).
- [9] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017, doi: [10.1109/ACCESS.2017.2730843](https://doi.org/10.1109/ACCESS.2017.2730843).
- [10] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, pp. 283–297, 2018, doi: [10.1016/j.scs.2018.02.014](https://doi.org/10.1016/j.scs.2018.02.014).
- [11] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "MedBlock: Efficient and secure medical data sharing via blockchain," *J. Med. Syst.*, vol. 42, Art. no. 136, 2018, doi: [10.1007/s10916-018-0993-7](https://doi.org/10.1007/s10916-018-0993-7).
- [12] Y. Chen, S. Ding, Z. Xu, H. Zheng, and S. Yang, "Blockchain-based medical records secure storage and medical service framework," *J. Med. Syst.*, vol. 43, Art. no. 5, 2019, doi: [10.1007/s10916-018-1121-4](https://doi.org/10.1007/s10916-018-1121-4).
- [13] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure EHRs sharing of mobile cloud based e-health systems," *IEEE Access*, vol. 7, pp. 66792–66806, 2019, doi: [10.1109/ACCESS.2019.2917555](https://doi.org/10.1109/ACCESS.2019.2917555).
- [14] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors*, vol. 19, no. 2, Art. no. 326, 2019, doi: [10.3390/s19020326](https://doi.org/10.3390/s19020326).
- [15] B. Zaabar, O. Cheikhrouhou, F. Jamil, M. Ammi, and M. Abid, "HealthBlock: A secure blockchain-based healthcare data management system," *Comput. Netw.*, vol. 200, Art. no. 108500, 2021, doi: [10.1016/j.comnet.2021.108500](https://doi.org/10.1016/j.comnet.2021.108500).
- [16] N. Rifi, E. Rachkidi, N. Agoulmine, and N. C. Taher, "Towards using blockchain technology for eHealth data access management," in *Proc. 4th Int. Conf. Adv. Biomed. Eng. (ICABME)*, pp. 1–4, 2017, doi: [10.1109/ICABME.2017.8167555](https://doi.org/10.1109/ICABME.2017.8167555).
- [17] Y. Zhang, D. He, and K.-K. R. Choo, "BaDS: Blockchain-based architecture for data sharing with ABS and CP-ABE in IoT," *Wireless Commun. Mobile Comput.*, vol. 2018, Art. no. 2783658, 2018, doi: [10.1155/2018/2783658](https://doi.org/10.1155/2018/2783658).
- [18] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO '85 (LNCS, vol. 218)*, pp. 417–426, 1986, doi: [10.1007/3-540-39799-X_31](https://doi.org/10.1007/3-540-39799-X_31).



- [19] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987, doi: [10.1090/S0025-5718-1987-0866109-5](https://doi.org/10.1090/S0025-5718-1987-0866109-5).
- [20] J. Benet, "IPFS—Content addressed, versioned, P2P file system," *arXiv preprint arXiv:1407.3561*, 2014, doi: [10.48550/arXiv.1407.3561](https://doi.org/10.48550/arXiv.1407.3561).
- [21] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, and Y. Alexandrov, "SmartCheck: Static analysis of Ethereum smart contracts," in *Proc. 1st Int. Workshop Emerg. Trends Softw. Eng. Blockchain (WETSEB)*, pp. 9–16, 2018, doi: [10.1145/3194113.3194115](https://doi.org/10.1145/3194113.3194115).
- [22] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, pp. 254–269, 2016, doi: [10.1145/2976749.2978309](https://doi.org/10.1145/2976749.2978309).
- [23] E. Androulaki et al., "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Art. no. 30, 2018, doi: [10.1145/3190508.3190538](https://doi.org/10.1145/3190508.3190538).