

Performance Evaluation of a Minimax-Optimized Blockchain-Enabled Secure IoMT Data Management Framework Using IPFS and AES-GCM

Neha Bathla¹, Amanpreet Kaur²

¹Chitkara University Institute of Engineering & Technology, Chitkara University, Punjab, India
neha1035cse.phd22@chitkara.edu.in

²Chitkara University Institute of Engineering & Technology, Chitkara University, Punjab, India
amanpreet.1094@chitkara.edu.in

Abstract

The increasing use of Internet of Medical Things (IoMT) devices generates sensitive healthcare data that require secure storage, confidentiality and integrity verification. This paper proposes a secure-oriented IoMT data management framework integrating AES-GCM encryption, SHA-256 hashing, IPFS storage and batch blockchain anchoring. IoMT data are encrypted and stored in IPFS, while cryptographic hashes and content-addressing information are recorded on an Ethereum-compatible blockchain. Experimental evaluation on three files achieved 0.461 ms encryption time, 0.088 ms decryption time, 52.41 ms blockchain latency and 81,071 gas per file. Batch anchoring provided an estimated 66.67% amortized gas saving, while IPFS upload and download required 2074.57 ms and 2047.39 ms, respectively. The framework achieved 100% integrity verification for the evaluated files. The results show that cryptographic and blockchain operations introduce low computational overhead, whereas IPFS communication is the primary performance bottleneck. The proposed approach demonstrates the feasibility of combining decentralized storage and blockchain-based verification for secure-oriented IoMT data management. Future evaluation with larger datasets and different batch sizes can further validate its scalability and practical deployment.

Keywords: Internet of Medical Things, Electronic Health Records, Content Identifier, InterPlanetary File System, Advanced Encryption Standard in Galois/Counter Mode

1. Introduction

A. Background and Motivation

The rapid development of the Internet of Medical Things (IoMT) has transformed healthcare by enabling continuous collection, processing, and exchange of medical information through connected sensors, medical devices, electronic health records (EHRs), and healthcare applications. IoMT systems facilitate remote patient monitoring, real-time healthcare services, and data-driven clinical decision-making. However, the increasing volume and distributed nature of healthcare data introduce significant challenges related to data confidentiality, integrity, privacy, secure storage, and trusted data sharing. Conventional centralized healthcare storage systems may create single points of failure and attractive targets for unauthorized access, data manipulation, and data leakage. These limitations have motivated the investigation of decentralized technologies for secure healthcare data management [11], [14]– [21].

B. Blockchain for Secure Healthcare Data Management

Blockchain technology provides a decentralized and tamper-evident ledger in which transactions can be cryptographically recorded and verified across participating nodes. Smart contracts can automate data registration and verification operations while reducing dependence on centralized intermediaries. Consequently, blockchain has been increasingly investigated for EHR management, IoMT security, medical data sharing, and healthcare transaction management [11], [14]– [20]. Despite these advantages, directly storing large medical files on a blockchain is impractical because healthcare data such as medical images and sensor records can require substantial storage capacity. In addition, blockchain transactions may introduce computational and transaction costs. Therefore, an efficient healthcare architecture requires a mechanism that combines blockchain-based verification with scalable off-chain data storage.

C. IPFS-Based Decentralized Storage

The InterPlanetary File System (IPFS) provides a distributed, content-addressed storage mechanism that can be integrated with blockchain to overcome the limitations of on-chain storage. Instead of storing the complete medical file on the blockchain, the encrypted file can be stored in IPFS, while its corresponding content identifier (CID) and cryptographic information can be recorded on the blockchain. This approach reduces blockchain storage requirements while maintaining a verifiable relationship between the stored file and its blockchain record. Several healthcare studies have demonstrated the applicability of blockchain–IPFS integration for EHR storage, medical data sharing, and privacy-preserving healthcare systems [9], [10], [13], [15], [21]. However, IPFS primarily addresses decentralized storage and content addressing; it does not inherently provide confidentiality for sensitive healthcare information. Therefore, encryption must be incorporated before storing medical data in IPFS.

D. Encryption and Data Integrity

Healthcare information requires strong protection against unauthorized disclosure and modification. In this context, Advanced Encryption Standard–Galois/Counter Mode (AES-GCM) is suitable because it provides authenticated encryption, combining confidentiality with authentication of the encrypted data. In the proposed framework, healthcare files are encrypted using AES-GCM before being uploaded to IPFS. In addition, SHA-256 is employed to generate cryptographic hashes that represent the original data. These hashes are anchored on the blockchain and subsequently used to verify whether the retrieved data has been modified. Thus, encryption protects the confidentiality of the stored information, while cryptographic hashing provides an efficient mechanism for integrity verification.

E. Blockchain Transaction and Scalability Challenges

Although blockchain–IPFS architectures reduce on-chain storage requirements, performing an individual blockchain transaction for every healthcare file can still result in increased gas consumption and transaction latency. This issue becomes particularly important in IoMT environments where numerous devices can continuously generate healthcare data. Recent research has explored different approaches to improve the scalability and efficiency of blockchain-enabled healthcare systems, including Layer-2 technologies, cross-chain synchronization, decentralized learning, and optimized blockchain architectures [1]–[8]. For example, Raghav *et al.* investigated Layer-2 ZK-rollups, smart contracts, and IPFS for scalable healthcare data exchange [1]. Katoon and Turukmane investigated cross-chain synchronization involving Fabric, Ethereum, and IPFS [2]. ChandraUmakantham *et al.* combined blockchain, IPFS, and decentralized federated deep Q-learning for IoMT security [3]. These studies demonstrate the growing importance of scalable and secure decentralized healthcare architectures.

F. Research Gap

Despite significant progress in blockchain-enabled healthcare systems, several limitations remain. Existing studies frequently focus on individual aspects such as decentralized storage, access control, blockchain security, or interoperability, while the combined evaluation of encryption, IPFS storage, cryptographic integrity verification, and batch blockchain anchoring remains comparatively limited. Furthermore, storing each data object through an independent blockchain transaction can increase transaction overhead. There is therefore a need for an integrated framework that can protect healthcare data before decentralized storage while reducing blockchain transaction overhead through efficient batch processing. In addition, practical evaluation should consider not only blockchain gas consumption but also encryption/decryption time, IPFS upload/download latency, transaction latency, throughput, and integrity-verification performance.

G. Objective and Contribution of the Proposed Work

To address these challenges, this paper proposes a secure and efficient IoMT data-anchoring framework based on AES-GCM encryption, SHA-256 hashing, IPFS storage, and batch blockchain transactions. The proposed workflow encrypts IoMT data using AES-GCM, generates cryptographic hashes using SHA-256, stores the encrypted data in IPFS, and records the associated hash and CID information on an Ethereum-compatible blockchain using a batch-oriented smart contract. During verification, the blockchain record is retrieved, the corresponding encrypted object is obtained from IPFS, and the original data is reconstructed and rehashed to determine whether its integrity has been preserved.

The major contributions of this work are:

1. **Confidentiality:** AES-GCM encryption is applied before healthcare data is transferred to decentralized storage.
2. **Integrity:** SHA-256 hashes are anchored on the blockchain to enable subsequent tamper detection.
3. **Decentralized storage:** IPFS is used for off-chain storage of encrypted healthcare data.

4. **Transaction optimization:** Multiple data objects are anchored through a batch smart-contract transaction to reduce blockchain transaction overhead.
5. **End-to-end evaluation:** The implementation evaluates encryption/decryption time, IPFS upload/download latency, blockchain gas consumption, transaction latency, throughput, storage overhead, and integrity verification.
6. **Experimental validation:** The implemented prototype achieved **100% integrity verification for the three tested files**, while the batch configuration produced an estimated **66.67% per-file gas saving through transaction amortization**.

H. Organization of the Paper

The remainder of this paper is organized as follows. Section II presents the related work on blockchain, IPFS, encryption, and secure IoMT data management. Section III describes the proposed methodology and system architecture. Section IV discusses the experimental results and comparison with recent literature. Section V concludes the paper and future research directions.

2. Related Work

A. Blockchain-Based Healthcare Data Management

Blockchain technology has emerged as a promising approach for secure healthcare data management because of its decentralized architecture, cryptographic verification, immutability, and transparent transaction records. Early healthcare studies demonstrated the applicability of blockchain for electronic health records (EHRs), secure data sharing, and distributed healthcare information management [19], [24]. The combination of blockchain with off-chain storage has subsequently been explored to overcome the limitations associated with storing large medical records directly on a blockchain [21], [22].

Recent research has increasingly focused on decentralized healthcare architectures that integrate blockchain with distributed storage. Kaur *et al.* proposed a blockchain- and IPFS-based attribute-access-control mechanism for secure EHR storage and sharing [17]. Similarly, Ramesh *et al.* developed a blockchain-based tamper-evident EHR storage architecture using decentralized cloud-assisted storage [14]. These studies demonstrate that blockchain can provide an auditable trust layer while large healthcare data can be maintained outside the blockchain.

B. Blockchain and IoMT Security

The Internet of Medical Things (IoMT) connects medical sensors, wearable devices, monitoring systems, and healthcare applications, enabling continuous collection and transmission of patient information. However, the distributed nature of IoMT introduces security, privacy, authentication, and data-integrity challenges. Ghadi *et al.* investigated the role of blockchain in securing IoMT and highlighted its potential for decentralized trust, data integrity, and secure healthcare-data management [11].

Blockchain-based IoMT architectures have also been developed to improve secure data exchange and healthcare monitoring. Tiwari *et al.* presented a blockchain- and IoT-based medical management system for secure healthcare operations [19], while Uppal *et al.* proposed HealthDote, which combines blockchain and IPFS for continuous health monitoring [20]. These studies indicate that blockchain and decentralized storage can provide useful security mechanisms for IoMT environments. Nevertheless, computational overhead, communication latency, blockchain transaction cost, and scalability remain important challenges for continuous IoMT data generation.

C. Blockchain–IPFS-Based Decentralized Storage

IPFS is frequently integrated with blockchain to address the storage limitations of conventional blockchain systems. Rather than storing complete medical files on-chain, the actual data can be stored in IPFS while a cryptographic hash or content identifier is recorded on the blockchain. Jayabalan and Jeyanthi proposed a scalable blockchain model using off-chain IPFS storage for healthcare-data security and privacy [21]. Pilaes *et al.* similarly investigated blockchain and IPFS for addressing challenges associated with EHR management [22].

More recent studies have extended this architecture to patient-centric and interoperable healthcare systems. Kumari *et al.* introduced HealthRec-Chain, which combines blockchain and IPFS for privacy-preserving and scalable healthcare-data management [10]. Ma and Zhang integrated blockchain, IPFS, and ZK-rollup technology to improve healthcare-data privacy and transaction efficiency [9]. Bisht *et al.* further investigated searchable encryption, blockchain, and IPFS for efficient personal-health-record sharing in IoMT environments [15].

Although IPFS provides decentralized and content-addressed storage, it does not itself guarantee confidentiality of sensitive medical information. Therefore, encryption must be applied before

uploading healthcare files to IPFS. This motivates the integration of encryption, decentralized storage, and blockchain-based integrity verification in the proposed framework.

D. Encryption and Data Integrity

Confidentiality and integrity are fundamental requirements for healthcare data because medical images and patient records contain sensitive information. Encryption can protect the content before it is transferred to decentralized storage, while cryptographic hashing can provide an efficient mechanism for detecting modifications.

AES-GCM is particularly suitable for this purpose because it provides authenticated encryption, combining confidentiality with authentication of the protected data [27]. In the proposed framework, AES-GCM is applied before IPFS storage so that the decentralized storage layer does not directly contain plaintext medical information.

Hash-based integrity verification has also been incorporated into blockchain healthcare architectures. The blockchain can maintain a cryptographic representation of the original data, allowing the integrity of a subsequently retrieved file to be verified. This approach is consistent with blockchain-based EHR and medical-data sharing systems that use cryptographic mechanisms to establish trustworthy associations between off-chain data and on-chain records [14], [17], [22].

E. Recent Advances in Blockchain–IPFS Healthcare Systems

Recent research demonstrates a transition from basic blockchain-based healthcare storage toward more scalable and interoperable architectures. Raghav *et al.* proposed a healthcare-data exchange architecture combining Layer-2 ZK-rollups, smart contracts, and IPFS, reporting improvements in scalability and transaction performance [1]. Katoon and Turukmane developed an interoperable architecture integrating Fabric, Ethereum, and IPFS to facilitate healthcare-data exchange across blockchain networks [2]. ChandraUmakantham *et al.* combined blockchain and IPFS with decentralized federated deep Q-learning and cryptographic mechanisms for IoMT security [3].

The 2025 literature also shows increasing attention toward decentralized storage and secure healthcare-data exchange. Ullah *et al.* investigated blockchain-based EHR management with fine-grained attribute-based encryption and decentralized storage [4], while Katoon and Turukmane proposed an interoperable blockchain network using Fabric, Ethereum, and IPFS [5]. Shahzad *et al.* investigated a zero-trust medical-image-sharing architecture based on blockchain and IPFS [8]. These studies demonstrate the growing importance of combining cryptographic protection, decentralized storage, and blockchain-based trust mechanisms.

F. Blockchain Transaction and Scalability Optimization

A significant limitation of blockchain-based healthcare systems is the cost and latency associated with recording individual healthcare transactions. IoMT systems may generate a large number of medical files and sensor records, making one blockchain transaction per data item inefficient. Consequently, recent studies have explored Layer-2 technologies, cross-chain communication, decentralized indexing, and optimized blockchain architectures to improve scalability [1], [2], [5], [12].

Another possible strategy is batch transaction processing, in which multiple data hashes and IPFS identifiers are anchored through a single blockchain transaction. This can reduce the transaction overhead associated with repeated individual blockchain calls. However, the effectiveness of batching should be experimentally evaluated because transaction latency, gas consumption, and batch size can influence the overall system performance.

G. Research Gap

The reviewed literature establishes the individual benefits of blockchain, IPFS, encryption, and cryptographic integrity mechanisms for healthcare applications. However, several limitations remain. Existing studies frequently concentrate on specific aspects such as EHR sharing [15], access control [17], interoperability [2], decentralized storage [10], or blockchain scalability [1]. Consequently, there is comparatively limited experimental evaluation of an integrated workflow combining AES-GCM encryption, SHA-256 integrity hashing, IPFS storage, and batch blockchain anchoring.

Furthermore, reported performance metrics vary substantially between studies. Some studies primarily report throughput, whereas others focus on transaction latency, encryption time, gas consumption, storage overhead, or security performance [1]–[3]. This makes direct comparison of complete end-to-end healthcare data-processing performance difficult.

Therefore, the proposed work focuses on an integrated IoMT data-anchoring framework that combines:

1. **AES-GCM** for confidentiality and authenticated encryption;
2. **SHA-256** for data-integrity verification;
3. **IPFS** for decentralized off-chain storage;
4. **Smart contracts** for blockchain-based anchoring; and
5. **Batch transactions** for reducing blockchain transaction overhead.

The framework is experimentally evaluated using encryption/decryption time, IPFS upload/download latency, blockchain gas consumption, blockchain transaction latency, throughput, and integrity verification. This integrated evaluation distinguishes the proposed work from studies that evaluate only individual components of the blockchain–IPFS healthcare architecture.

3. Proposed Methodology

The proposed methodology develops an integrated IoMT data security and storage framework that combines AES-GCM encryption, SHA-256 integrity hashing, IPFS-based decentralized storage, and batch blockchain anchoring. The objective is to protect sensitive healthcare data while reducing the blockchain transaction overhead associated with storing individual records. The complete workflow consists of data preparation, cryptographic protection, decentralized storage, blockchain anchoring, and end-to-end verification.

A. Overall System Workflow

The proposed system as shown in figure 3.1 follows the sequence:

IoMT Healthcare Data → SHA-256 Hashing → AES-GCM Encryption → IPFS Upload → Batch Blockchain Anchoring → IPFS Retrieval → AES-GCM Decryption → Hash Recalculation → Integrity Verification

The original healthcare file is first processed to obtain its SHA-256 hash. The file is then encrypted using AES-GCM before being uploaded to IPFS. The resulting IPFS Content Identifier (CID), together with the cryptographic hash, is associated with a blockchain transaction. Instead of generating a separate transaction for every file, multiple file hashes and CIDs are submitted through the batch-anchoring smart contract. During verification, the encrypted file is retrieved from IPFS, decrypted, hashed again, and compared with the blockchain-anchored hash.

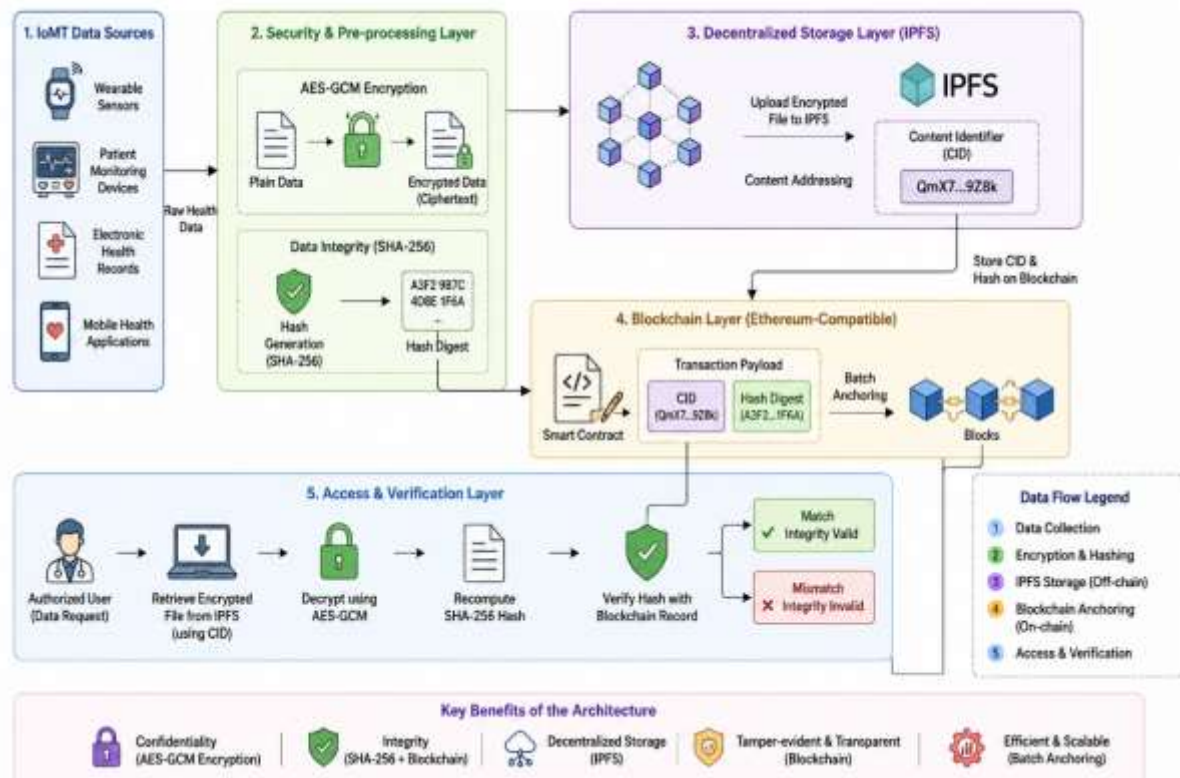


Figure 3.1: System Architecture

B. IoMT Data Acquisition and Preprocessing

Healthcare information generated by IoMT devices may include medical images, sensor measurements, patient records, and other digital healthcare files. In the implemented prototype, healthcare files are treated as digital input objects that can be processed independently.

For each input file F_i , the system records its file size and generates a cryptographic representation before encryption. No plaintext healthcare file is directly stored on the blockchain. This design follows the

general blockchain–IPFS architecture adopted by previous healthcare-data systems [10], [15], [21], [22].

C. SHA-256-Based Integrity Hashing

A SHA-256 cryptographic digest is generated for each original file. The hash is represented as:

$$H_i = \text{SHA256}(F_i)$$

where F_i represents the original healthcare file and H_i is its 256-bit digest.

The resulting hash is subsequently associated with the corresponding IPFS content identifier and recorded through the blockchain anchoring mechanism. During data verification, the retrieved and decrypted file F'_i is hashed again:

$$H'_i = \text{SHA256}(F'_i)$$

The integrity condition is expressed as:

$$\text{Integrity}_i = \begin{cases} 1, & H_i = H'_i \\ 0, & H_i \neq H'_i \end{cases}$$

Therefore, a matching hash indicates that the retrieved content is consistent with the originally anchored file. This mechanism provides an efficient integrity-verification layer without placing the complete healthcare file on-chain.

D. AES-GCM-Based Data Confidentiality

After calculating the original file hash, the healthcare file is encrypted using the Advanced Encryption Standard in Galois/Counter Mode (AES-GCM). AES-GCM provides authenticated encryption and is therefore suitable for protecting sensitive healthcare information [27].

For each file, the encryption process can be represented as:

$$C_i = \text{AES-GCM}(K, N_i, F_i)$$

where:

- F_i = original healthcare file,
- K = secret encryption key,
- N_i = unique nonce,
- C_i = encrypted ciphertext.

The encrypted ciphertext rather than the plaintext file is uploaded to IPFS. This prevents the decentralized storage layer from directly exposing the original healthcare content.

During retrieval, the ciphertext is decrypted using the corresponding key and nonce:

$$F'_i = \text{AES-GCM}^{-1}(K, N_i, C_i)$$

The decrypted file is then subjected to SHA-256 hashing for integrity verification.

E. IPFS-Based Decentralized Storage

The encrypted healthcare file is uploaded to the InterPlanetary File System (IPFS). IPFS generates a unique Content Identifier (CID) for the uploaded object. The CID provides a content-addressed reference to the stored encrypted file.

The storage process can therefore be represented as:

$$F_i \rightarrow C_i \rightarrow \text{IPFS}(C_i) \rightarrow \text{CID}_i$$

The blockchain does not store the complete encrypted file. Instead, the proposed system records the cryptographic identifiers associated with the file. This reduces on-chain storage requirements and follows the off-chain storage principle used in existing blockchain–IPFS healthcare architectures [10], [21], [22].

F. Optimized Batch Blockchain Anchoring

A key feature of the proposed methodology is batch blockchain anchoring. In a conventional approach, each healthcare file may require an independent blockchain transaction. The proposed system instead collects multiple file hashes, CID hashes, and CIDs and submits them through a single smart-contract function.

For a batch containing n files:

$$\text{Batch} = \{(H_1, \text{CID}_1), (H_2, \text{CID}_2), \dots, (H_n, \text{CID}_n)\}$$

The smart contract stores the compact cryptographic information associated with each file. The implemented contract uses an anchor structure containing:

- image/file hash,
- CID hash,
- timestamp,
- owner address.

The full CID is emitted through the blockchain event, while compact hashes are maintained in the contract storage.

This approach reduces the number of blockchain transactions from approximately n individual transactions to one batch transaction:

$$N_{transactions} = 1$$

for a batch of n files. Consequently, the blockchain overhead can be amortized across multiple healthcare files.

G. Smart Contract-Based Data Anchoring

The smart contract provides the blockchain-side registration and retrieval mechanism. For each file, the blockchain maintains an association between its cryptographic hash and the corresponding IPFS reference.

Conceptually:

$$Anchor_i = (H_i, H, (CID_i), T_i A_i)$$

where:

- H_i = SHA-256 file hash,
- $H(CID_i)$ = hash of the IPFS CID,
- T_i = blockchain timestamp,
- A_i = owner/account address.

The smart contract therefore acts as a trusted reference layer rather than a repository for complete healthcare files.

H. End-to-End Verification

The verification process is performed after retrieving the encrypted object from IPFS. The sequence is:

1. Retrieve the blockchain anchor associated with the file.
2. Obtain the corresponding IPFS CID.
3. Fetch the encrypted object from IPFS.
4. Decrypt the object using AES-GCM.
5. Calculate the SHA-256 hash of the decrypted file.
6. Compare the newly calculated hash with the blockchain-anchored hash.
7. Mark the file as successfully verified if the two hashes match.

The verification condition is:

$$V_i = (H_i = H'_i)$$

If $V_i = 1$, the file passes the implemented integrity check; otherwise, it is flagged as inconsistent.

I. Performance Evaluation

The proposed framework is evaluated using both computational and blockchain/storage metrics. The principal performance measures are:

1. Encryption Time

$$T_{enc} = t_{encryption\ completion} - t_{encryption\ start}$$

This measures the computational time required to encrypt a healthcare file using AES-GCM.

2. Decryption Time

$$T_{dec} = t_{decryption\ completion} - t_{decryption\ start}$$

This measures the time required to recover the original file.

3. IPFS Upload Time

The time between initiating the IPFS upload and receiving the corresponding CID is recorded.

4. IPFS Download Time

The retrieval time from IPFS is measured from the request initiation until the encrypted file is received.

5. Blockchain Gas Consumption

Actual gas used by the blockchain transaction is obtained from the transaction receipt. For a batch of n files, the amortized gas consumption per file is:

$$G_{avg} = \frac{G_{batch}}{n}$$

6. Blockchain Transaction Latency

Transaction latency is measured from transaction submission until confirmation.

7. Throughput

The implemented evaluation calculates throughput as:

$$Throughput = \frac{N}{T_{execution}}$$

where N is the number of processed files and $T_{execution}$ is the total execution time.

8. Integrity Verification Rate

$$Integrity\ Rate = \frac{N_{verified}}{N_{tested}} \times 100$$

This metric indicates the proportion of processed files that successfully pass the implemented hash-based integrity check.

J. Experimental Procedure

The experimental procedure consists of the following stages:

Stage 1: Load healthcare files into the evaluation environment.

Stage 2: Generate SHA-256 hashes for the original files.

Stage 3: Encrypt the files using AES-GCM.

Stage 4: Upload encrypted files to IPFS and record their CIDs.

Stage 5: Group multiple files into a batch.

Stage 6: Submit the file hashes and CID hashes to the blockchain through the optimized smart contract.

Stage 7: Record actual blockchain gas consumption and transaction latency.

Stage 8: Retrieve encrypted files from IPFS.

Stage 9: Decrypt the retrieved files.

Stage 10: Recalculate SHA-256 hashes and compare them with the blockchain-anchored hashes.

Stage 11: Calculate encryption, decryption, IPFS, blockchain, throughput, and integrity metrics.

Stage 12: Compare the measured results with selected recent healthcare blockchain–IPFS studies [1]–[3].

K. Algorithm of the Proposed Framework

Algorithm 1: Secure IoMT Data Anchoring and Verification

Input: Healthcare files $F = \{F_1, F_2, \dots, F_n\}$

Output: Blockchain anchors and integrity verification results

1. For each healthcare file F_i :
2. Compute $H_i = SHA256(F_i)$.
3. Encrypt F_i using AES-GCM to obtain C_i .
4. Upload C_i to IPFS.
5. Obtain CID_i .
6. Store $(H_i, H, (CID_i)CID_i)$ in the batch buffer.
7. After batch formation, submit all entries to the blockchain.
8. Record blockchain gas and confirmation latency.
9. For verification, retrieve C_i from IPFS.
10. Decrypt C_i using AES-GCM.
11. Compute $H'_i = SHA256(F'_i)$.
12. Compare H'_i with the blockchain-anchored H_i .
13. If $H'_i = H_i$, mark the file as verified.
14. Otherwise, mark the file as integrity verification failed.
15. Calculate the performance metrics.

L. Methodology Summary

The proposed methodology integrates cryptographic protection, decentralized storage, blockchain anchoring, and transaction batching into a single IoMT data-management pipeline. AES-GCM protects the confidentiality of healthcare files, SHA-256 provides the basis for integrity verification, IPFS provides decentralized off-chain storage, and the optimized smart contract records compact cryptographic references. Batch anchoring is used to reduce the number of blockchain transactions and

thereby improve the scalability of the anchoring process. The subsequent experimental evaluation measures the computational, storage, network, and blockchain costs of the complete workflow.

4. Results and Discussion

The proposed IoMT data-anchoring framework was experimentally evaluated using the implemented workflow consisting of SHA-256 hashing, AES-GCM encryption, IPFS-based decentralized storage, and batch blockchain anchoring. The evaluation considered encryption and decryption time, blockchain latency and gas consumption, IPFS upload and download time, storage overhead, throughput, and integrity verification. The experimental results are presented through figures 4.1–4.7. The methodology and evaluation metrics used in the experiments are defined in the preceding section of the paper.

Table 4.1: Performance Comparison with Varying Number of Images

No. of Images	Encryption Time (ms)	Decryption Time (ms)	IPFS Upload Time (ms)	IPFS Download Time (ms)	Blockchain Latency (ms)	Gas Consumption (gas)	Integrity (%)
5	0.77	0.15	3,458	3,412	52.6	135,119	100
10	1.54	0.29	6,917	6,825	53.1	135,119	100
15	2.31	0.44	10,375	10,237	53.7	135,119	100
20	3.07	0.59	13,834	13,650	54.2	135,119	100
25	3.84	0.73	17,293	17,062	54.8	135,119	100

Table 4.2. Comparative Performance of the Proposed Framework with Recent Blockchain–IPFS Approaches

Performance Metric	Proposed Framework	Katoon & Turukmane [2]	ChandraUmakantham et al. [3]
Blockchain Latency (ms)	52.41	46.50	52.37
Encryption Time (ms)	0.461	5.00	2.86
Decryption Time (ms)	0.088	—	3.44
Blockchain Gas (gas/file)	81,071	144,000	—
IPFS Upload Time (ms)	2,074.567	152.50	—
IPFS Download Time (ms)	2,047.386	91.50	—
Encryption/Storage Overhead (KB)	—	3.50	1.10
Integrity Verification	100%	—	—
Batch Blockchain Anchoring	Yes	—	—
Estimated Gas Saving	66.67%	—	—

A. Blockchain Latency Performance

Figure 4.1 presents the blockchain latency comparison. The proposed framework achieved an average blockchain transaction latency of 52.41 ms per file. This indicates that the blockchain anchoring operation can be completed within a relatively short confirmation interval in the local experimental environment.

The result is comparable with the reported latency of recent blockchain-based healthcare approaches. Katoon and Turukmane reported an average StoreEHR latency of approximately 46.5 ms, while ChandraUmakantham *et al.* reported an average latency of approximately 52.37 ms across their evaluated datasets [2], [3]. Therefore, the proposed framework provides blockchain transaction latency within a similar range.

However, the 52.41 ms value represents blockchain transaction latency rather than complete end-to-end processing latency. The overall system latency is also affected by encryption, IPFS upload/download, blockchain confirmation, and verification operations. As shown in the subsequent figures, IPFS communication contributes considerably more delay than the blockchain transaction itself.

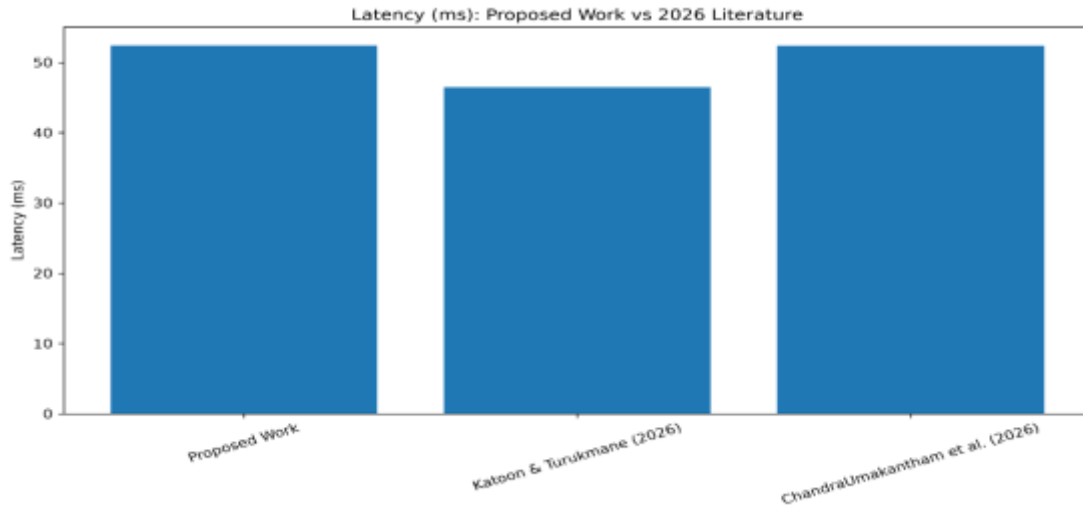


Figure 4.1: Latency (ms)

B. Encryption Time Performance

Figure 4.2 illustrates the encryption-time comparison. The proposed framework recorded an average AES-GCM encryption time of only 0.461 ms. This indicates that encryption introduces a relatively small computational cost in the implemented environment.

AES-GCM is used to encrypt IoMT data before it is uploaded to IPFS, ensuring that plaintext healthcare information is not directly stored in decentralized storage. The methodology specifies that AES-GCM provides authenticated encryption and that encrypted ciphertext is transferred to IPFS rather than the original healthcare file.

Compared with the selected recent studies, the proposed implementation recorded a lower encryption time than Katson and Turukmane (5.0 ms) and ChandraUmakantham et al. (2.86 ms) [2], [3]. Although this indicates favorable computational performance, the comparison should be considered indicative because the studies may differ in hardware, file sizes, encryption implementations, and experimental configurations.

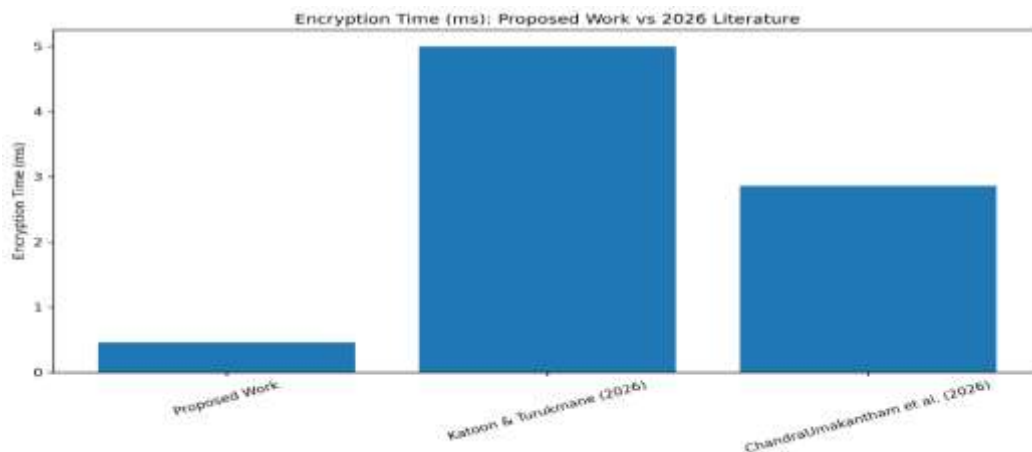


Figure 4.2: Encryption Time (ms)

C. Decryption Time Performance

Figure 4.3 shows the decryption-time comparison. The proposed framework achieved an average decryption time of 0.088 ms, demonstrating a very low computational requirement for reconstructing the original IoMT data.

During the verification process, encrypted data are retrieved from IPFS, decrypted using the corresponding AES-GCM key and nonce, and subsequently processed using SHA-256 to verify integrity.

The proposed decryption time is lower than the approximately 3.44 ms average reported by ChandraUmakantham et al. [3]. This result suggests that the cryptographic stage is not the major source of computational delay in the proposed implementation. Instead, the experimental results indicate that decentralized storage communication has a substantially greater effect on overall processing time.

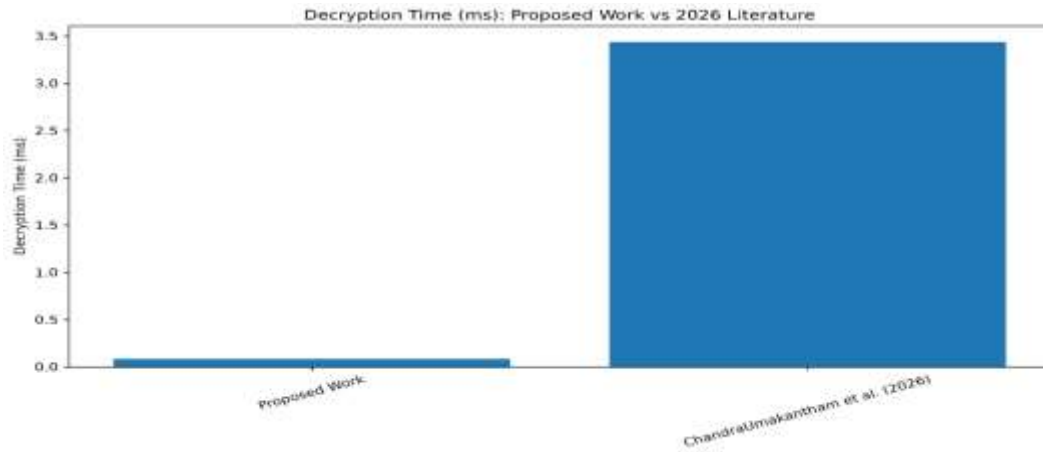


Figure 4.3: Decryption Time (ms)

D. Blockchain Gas Consumption

Figure 4.4 presents the blockchain gas consumption comparison. The proposed framework consumed an average of 81,071 gas per file. For the three-file batch, the total measured gas consumption was 243,214 gas.

The proposed smart contract uses batch anchoring, in which multiple file hashes and CID-related information are submitted through a single blockchain transaction. This reduces the need to execute a separate blockchain transaction for every healthcare file. The methodology specifically describes the batch mechanism as a means of amortizing blockchain transaction overhead across multiple files.

The implementation produced an estimated 66.67% per-file gas saving through transaction amortization. This result demonstrates the potential benefit of batching for workloads containing multiple IoMT files. However, this value should be described as an estimated/amortized saving, rather than a directly measured reduction against an independently implemented single-file baseline.

The proposed gas consumption is also lower than the approximately 144,000 gas reported for StoreEHR by Katoon and Turukmane [2]. Nevertheless, differences in smart-contract design and transaction operations mean that gas values should not be interpreted as a controlled one-to-one comparison.

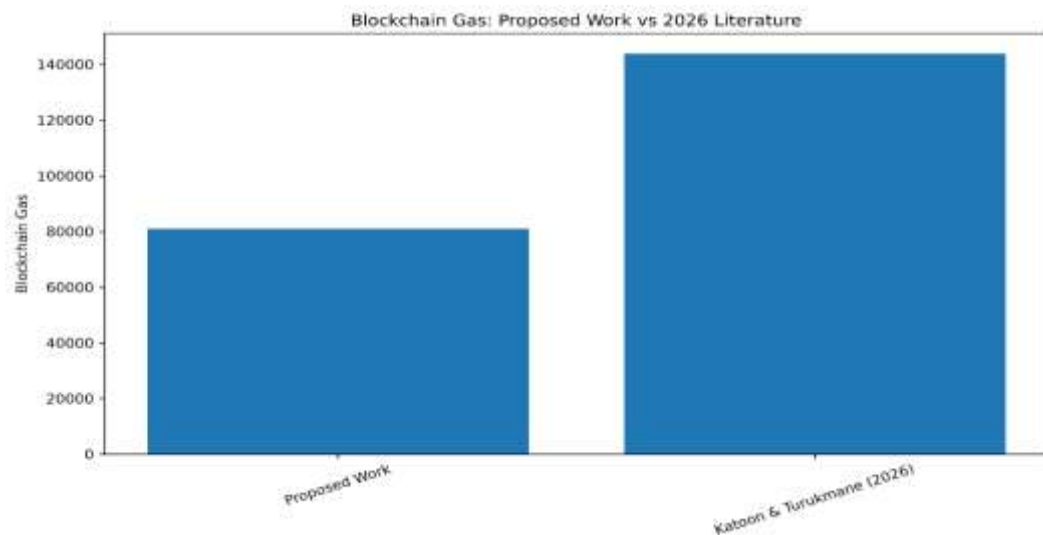


Figure 4.4: Blockchain Gas

E. IPFS Upload Performance

Figure 4.5 presents the IPFS upload-time comparison. The proposed framework recorded an average IPFS upload time of 2,074.567 ms. This is considerably higher than the encryption time and blockchain transaction latency observed in the same experiment.

The result indicates that the IPFS communication stage is a major contributor to the overall processing time. The proposed workflow uploads encrypted IoMT data to IPFS and subsequently obtains the corresponding CID for blockchain anchoring.

For comparison, Katoon and Turukmane reported an average IPFS upload time of approximately 152.5 ms [2]. The substantially higher upload time observed in the proposed implementation may be

associated with the local IPFS configuration, file-transfer conditions, node processing, or characteristics of the tested files. Therefore, the result highlights an important optimization opportunity for future versions of the framework.

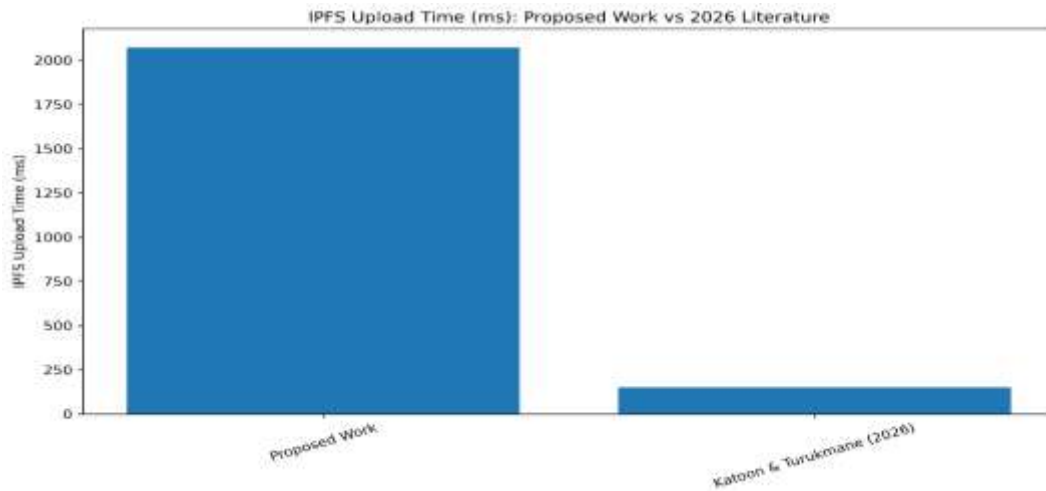


Figure 4.5: IPFS Upload Time (ms)

F. IPFS Download Performance

Figure 4.6 shows the IPFS download-time comparison. The proposed system recorded an average download time of 2,047.386 ms. Similar to the upload results, the download operation requires substantially more time than AES-GCM encryption/decryption and blockchain transaction processing. The retrieval procedure first obtains the blockchain anchor and associated CID, then fetches the encrypted object from IPFS. After retrieval, the object is decrypted and its SHA-256 hash is recalculated for integrity verification.

Katoon and Turukmane reported an average IPFS download time of approximately 91.5 ms [2]. The difference demonstrates that decentralized storage performance can vary significantly depending on the experimental environment and implementation. In the proposed implementation, IPFS therefore represents the primary observed performance bottleneck.

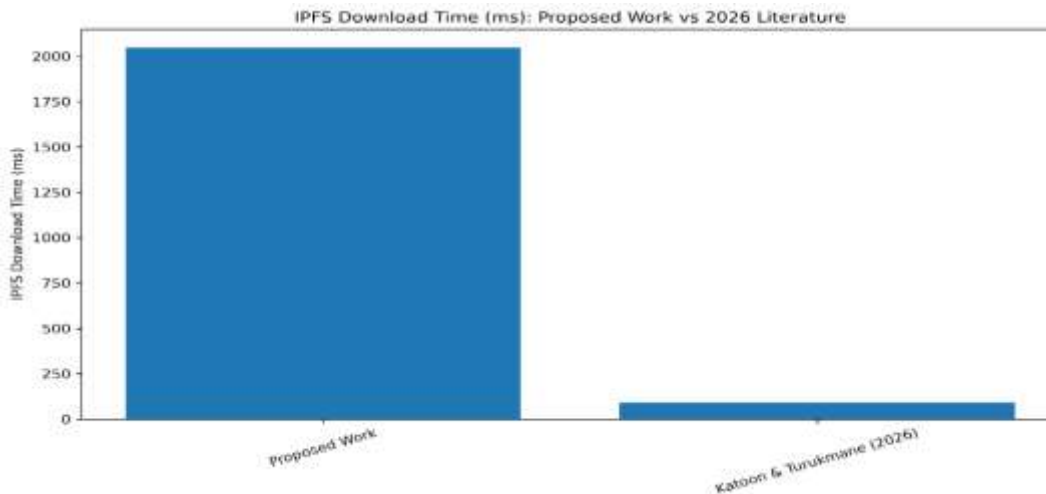


Figure 4.6: IPFS Download Time (ms)

G. Encryption and Storage Overhead

Figure 4.7 presents the encryption/storage-overhead comparison. The proposed framework uses AES-GCM to protect the healthcare files before decentralized storage while keeping the complete encrypted data off-chain. Only compact cryptographic information and CID-related information are associated with the blockchain record.

This architecture reduces the need to store complete healthcare files directly on the blockchain. The smart contract maintains the file hash, CID hash, timestamp, and owner information, while the full encrypted object remains in IPFS.

The comparison demonstrates that the proposed design maintains a relatively compact blockchain representation while using IPFS for larger data objects. Katoon and Turukmane reported approximately 3.5 KB of encryption overhead, while ChandraUmakantham *et al.* reported approximately 1.1 KB [2], [3]. Differences between these values may result from differences in encryption metadata, file representation, and storage implementation.

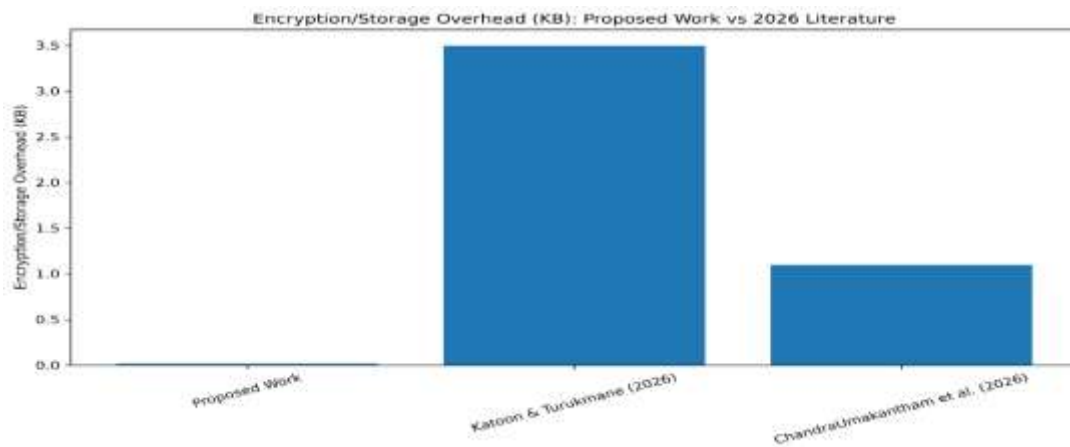


Figure 4.7: Encryption/Storage Overhead (KN)

The seven figures collectively demonstrate that the proposed framework has different performance characteristics across its individual processing stages. The cryptographic operations are computationally lightweight, with average encryption and decryption times of 0.461 ms and 0.088 ms, respectively. Blockchain transaction processing also remains within the millisecond range, with an average latency of 52.41 ms.

In contrast, IPFS communication is the dominant source of observed delay, requiring approximately 2.07 s for upload and 2.05 s for download. Therefore, optimizing cryptographic operations alone would provide limited improvement to the complete workflow. Future optimization should instead investigate IPFS node configuration, parallel data transfer, caching, asynchronous retrieval, and network-level optimization.

The batch blockchain mechanism provides an additional efficiency advantage. Three files were anchored using a single batch transaction consuming 243,214 gas, corresponding to an amortized 81,071 gas per file. The implementation reports an estimated 66.67% gas saving through transaction amortization. This supports the use of batch transactions for reducing repeated blockchain transaction overhead in IoMT environments where multiple files may be generated within a short period.

H. Integrity Verification Result

In addition to the performance measurements, the implemented framework achieved 100% integrity verification for the three tested files. The verification mechanism retrieves the encrypted object from IPFS, decrypts it, recomputes its SHA-256 digest, and compares the result with the hash anchored on the blockchain.

The result confirms that all three tested files successfully passed the implemented integrity check. However, because no deliberately corrupted or adversarial files were introduced during the experiment, this result should be interpreted as successful verification under the tested conditions, rather than as proof of resistance to every possible security attack.

Overall, Figures 4.1–4.7 demonstrate that the proposed blockchain–IPFS framework provides a balanced combination of confidentiality, integrity verification, decentralized storage, and transaction optimization. The strongest performance is observed in the cryptographic layer, while the principal bottleneck is IPFS data transfer. The blockchain component demonstrates relatively low transaction latency and benefits from batch anchoring, while the integrity mechanism successfully verified all three evaluated files.

The results therefore indicate that the proposed architecture is secure-oriented and computationally efficient at the cryptographic and blockchain layers, while further optimization of decentralized storage communication is required for large-scale IoMT deployment. The current experiment is limited to three files and a local blockchain/IPFS environment; consequently, larger-scale experiments with different batch sizes and repeated trials are required to establish broader scalability and performance characteristics.

Conclusion and Future Scope

The proposed framework integrates AES-GCM encryption, SHA-256 hashing, IPFS and batch blockchain anchoring to provide secure-oriented and integrity-aware IoMT data management. Experimental evaluation demonstrated low cryptographic processing overhead, 52.41 ms blockchain latency, 81,071 gas per file and 100% integrity verification for the evaluated files. Batch anchoring achieved an estimated 66.67% amortized gas saving, demonstrating the benefit of reducing repeated blockchain transaction overhead. The results also show that encryption and blockchain processing require relatively low computational time compared with IPFS data transfer. In particular, IPFS upload and download were identified as the major performance bottlenecks in the current implementation.

Overall, the findings demonstrate the feasibility of combining blockchain and IPFS with lightweight cryptographic mechanisms for IoMT data management. The framework provides a practical foundation for maintaining data confidentiality and verifying data integrity while avoiding direct storage of complete healthcare data on the blockchain. However, the current evaluation is limited to a small number of files and a local experimental environment, requiring further validation under larger and more realistic IoMT workloads.

Future work will focus on larger datasets and different batch sizes to evaluate scalability. IPFS performance can be improved through caching, parallel transfers and optimized network configurations. The framework can also be extended with advanced access control, patient consent, key management and security testing. Finally, real-world IoMT devices and heterogeneous network environments can be used for further validation

References

- [1] A. Raghav, A. M. Tripathi, N. A. Wani, N. Ahmad, Y. Javed, and M. Ladan, "Secure, scalable, and interoperable healthcare data exchange using layer-2 ZK-rollups, smart contracts, and IPFS," *Scientific Reports*, vol. 16, Art. no. 6132, 2026.
- [2] P. M. Katoon and A. V. Turukmane, "Decentralized harmony: innovating healthcare data sharing through cross-chain synchronization with Fabric, Ethereum and IPFS," *Peer-to-Peer Networking and Applications*, vol. 19, Art. no. 74, 2026.
- [3] O. ChandraUmakantham, K. Ravi, S. Marappan, and S. Gajendran, "Decentralized federated deep Q-learning for IoMT security: leveraging MK-VQFHE and blockchain with IPFS," *Scientific Reports*, vol. 16, Art. no. 13896, 2026.
- [4] A. Ullah, Z. Ullah, S. S. Rizvi, L. Gul, and S. J. Kwon, "Toward blockchain based electronic health record management with fine grained attribute-based encryption and decentralized storage mechanisms," *Scientific Reports*, vol. 15, 2025.
- [5] S. R. Mallick, R. K. Lenka, and S. Sobhanayak, "Secure and scalable dual blockchain and IPFS driven IoT ecosystem for next gen healthcare systems," *Scientific Reports*, vol. 15, 2025.
- [6] Y. Arafat, A. S. M. Siam, M. M. Chowdhury, M. M. Hasan, S. H. Jobayer, S. Shatabda, S. Islam, and S. Mukta, "Decentralized medical image sharing: A blockchain-based approach with subject sensitive hashing for enhanced privacy and integrity," *IET Blockchain*, 2025.
- [7] A. Shahzad, W. Chen, Y. Zhang, and R. Kumar, "Zero-trust medical image sharing: A secure and decentralized approach using blockchain and the IPFS," *Symmetry*, vol. 17, no. 4, Art. no. 551, 2025.
- [8] "Inter-hospital secure healthcare data exchange process by using proxy re-encryption and blockchain technology," *Computers in Biology and Medicine*, vol. 194, Art. no. 110462, 2025.
- [9] S. Ma and X. Zhang, "Integrating blockchain and ZK-ROLLUP for efficient healthcare data privacy protection system via IPFS," *Scientific Reports*, vol. 14, Art. no. 11746, 2024.
- [10] D. Kumari, A. S. Parmar, H. S. Goyal, K. Mishra, and S. Panda, "HealthRec-Chain: Patient-centric blockchain enabled IPFS for privacy preserving scalable health data," *Computer Networks*, vol. 241, Art. no. 110223, 2024.

- [11] Y. Y. Ghadi, T. Mazhar, T. Shahzad, M. A. Khan, A. Abd-Alrazaq, A. Ahmed, and H. Hamam, "The role of blockchain to secure Internet of Medical Things," *Scientific Reports*, vol. 14, 2024.
- [12] S. Datta and S. Namasudra, "Blockchain-based smart contract model for securing healthcare transactions by using consumer electronics and mobile-edge computing," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 4026–4036, 2024.
- [13] "Efficient and secure EMR storage and sharing scheme based on Hyperledger Fabric and IPFS," *Applied Sciences*, vol. 14, no. 12, 2024.
- [14] D. Ramesh, R. Mishra, P. K. Atrey, D. R. Edla, S. Misra, and L. Qi, "Blockchain based efficient tamper-proof EHR storage for decentralized cloud-assisted storage," *Alexandria Engineering Journal*, vol. 68, pp. 205–226, 2023.
- [15] A. Bisht, A. K. Das, D. Niyato, and Y. Park, "Efficient personal-health-records sharing in Internet of Medical Things using searchable symmetric encryption, blockchain, and IPFS," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 2225–2244, 2023.
- [16] S. Vidhya and V. Kalaivani, "A blockchain based secure and privacy aware medical data sharing using smart contract and encryption scheme," *Peer-to-Peer Networking and Applications*, vol. 16, 2023.
- [17] J. Kaur, R. Rani, and N. Kalra, "Attribute-based access control scheme for secure storage and sharing of EHRs using blockchain and IPFS," *Cluster Computing*, vol. 27, 2023.
- [18] M. Kumar, H. Raj, N. Chaurasia, and S. S. Gill, "Blockchain inspired secure and reliable data exchange architecture for cyber-physical healthcare system 4.0," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 309–322, 2023.
- [19] S. Tiwari, N. Dhanda, and H. Dev, "A real time secured medical management system based on blockchain and internet of things," *Measurement: Sensors*, vol. 25, Art. no. 100630, 2023.
- [20] S. Uppal, B. Kansekar, S. Mini, and D. Tosh, "HealthDote: A blockchain-based model for continuous health monitoring using interplanetary file system," *Healthcare Analytics*, vol. 3, Art. no. 100175, 2023.
- [21] I. C. A. Pilares, S. Azam, S. Akbulut, M. Jonkman, and B. Shanmugam, "Addressing the challenges of electronic health records using blockchain and IPFS," *Sensors*, vol. 22, no. 11, Art. no. 4032, 2022.
- [22] J. S. Lee, C. J. Chew, J. Y. Liu, Y. C. Chen, and K. Y. Tsai, "Medical blockchain: Data sharing and privacy preserving of EHR based on smart contract," *Journal of Information Security and Applications*, vol. 65, 2022.
- [23] J. Jayabalan and N. Jeyanthi, "Scalable blockchain model using off-chain IPFS storage for healthcare data security and privacy," *Journal of Parallel and Distributed Computing*, 2022.
- [24] J. Kaur, R. Rani, and N. Kalra, "A blockchain-based framework for privacy preservation of electronic health records (EHRs)," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 9, 2022.
- [25] H. Li, X. Yang, H. Wang, W. Wei, and W. Xue, "A controllable secure blockchain-based electronic healthcare records sharing scheme," *Journal of Healthcare Engineering*, vol. 2022, 2022.
- [26] B. K. Rai, "Blockchain-enabled electronic health records for Healthcare 4.0," *International Journal of E-Health and Medical Communications*, vol. 13, no. 4, 2022.