International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(18s) 2026 pp. 613-626
https://doi.org/10.70917/ijcisim-2026-4890
Article

ENHANCED CRYPTOGRAPHIC INTEGRITY: INTEGRATING MODIFIED SHA-256 FOR SECURE DISTRIBUTED LEDGER-BASED MEDICAL RECORD SYSTEMS

R. Palaniyappan[1] , D. Sasikala[2]

[1]Department of Computer Science, Sri Vasavi College, Erode, Tamilnadu, India.
palaniraja92@gmail.com
[2]Department of Computer Science, Sri Vasavi College, Erode, Tamilnadu, India.
drdsasikalaphd@gmail.com

Abstract: The integration of a Modified SHA-256 algorithm into distributed ledger-based medical record systems enhances cryptographic security, data integrity, and computational efficiency. The proposed model optimizes padding, message scheduling, and compression functions. A public repository containing 50,000 de-identified Electronic Health Records (EHRs) was utilized for evaluation under a Python-based AWS cloud environment with Intel Core i9 (3.6 GHz), 32 GB RAM, and NVIDIA RTX 3080 GPU. Results demonstrate that the Modified SHA-256 outperforms Standard SHA-256 and related models, reducing hash computation time by 33.6%, latency by 35.6%, encryption/decryption time by 28.2%, and storage overhead by 31.4% on average. The system also lowers computational complexity and energy consumption without compromising cryptographic robustness. Integration with distributed ledger ensures immutability, Merkle tree validation, and verification digest mechanisms, providing real-time consistency. Overall, the proposed Modified SHA-256-based distributed ledger framework establishes a secure and efficient foundation for healthcare data management and privacy preservation..
Keywords: Medical record, SHA-256, data integrity, HER, Markle Tree, Digest Mechanism, Privacy Preservation

1. Introduction

The management and protection of medical records have become a critical concern in modern healthcare ecosystems due to the increasing digitization of patient information and the growing reliance on cloud-based medical data systems [1]. Traditional centralized Electronic Health Record (EHR) architectures, though widely adopted, face substantial challenges in ensuring security, privacy, and integrity [2]. Centralized repositories are prone to single points of failure, data tampering, cyberattacks, and unauthorized access. Moreover, the lack of interoperability across heterogeneous healthcare systems complicates secure data sharing among hospitals, laboratories, and insurance providers [3]. As medical records often include highly sensitive information such as patient identities, diagnoses, prescriptions, and medical imaging, the inability to guarantee immutability and verifiable authenticity exposes healthcare infrastructures to significant risks, including data breaches, fraudulent modifications, and privacy violations [4].

To overcome these limitations, distributed ledger technology has been introduced as a decentralized and immutable ledger that ensures transparent and secure storage of medical data [5]. In a distributed ledger-based healthcare model, each transaction or record is encrypted, time-stamped, and linked to the previous block, thereby preventing unauthorized alterations. However, while distributed ledger inherently strengthens integrity and traceability, its efficiency depends largely on the underlying cryptographic hashing algorithm [6]. The Secure Hash

Algorithm 256 (SHA-256), which serves as the cryptographic backbone of many distributed ledger frameworks, provides strong resistance against preimage and collision attacks but introduces significant computational overhead. This overhead leads to increased latency, higher power consumption, and reduced scalability, which are major bottlenecks in healthcare systems that demand real-time access, high transaction throughput, and lightweight processing for IoMT devices [7].

Adding a modified SHA256 algorithm to distributed ledger based medical record systems greatly increases security, integrity, and efficiency in the handling of patient data [8]. Though SHA256 is used for cryptographic security, it has computational overheads that can be optimized for medical applications. The proposed modifications include optimized preprocessing, stronger compression functions, and light-weight cryptographic operations achieving faster and more efficient hashing while maintaining robust security guarantees [9]. Sensitive information such as diagnoses, prescriptions, and imaging data that is compiled into the medical records need to be stored in an immutable and verifiable storage to prevent unauthorized modifications and keep patients' private [10].

The modified SHA256 algorithm optimizes padding, message scheduling and compression function to improve hashing performance [11]. The system serializes patient records into a binary format before applying hashing by structuring patient records in interoperable formats such as JSON [12]. The modified algorithm ensures collision resistance yet at the same time makes hash computations lightweight and cuts down verification time and computational costs [13].

The hashed records are secured due to distributed ledger integration by embedding them inside blocks, each containing a timestamp and a reference to the previous block's hash. This structure ensures data integrity, which is computationally infeasible to tamper. More security is added by verification layers that use Merkle tree-based validation to provide efficient proof of integrity. Hashes are recomputed to allow real time verification of medical records to verify consistency and authenticity dispersed across multiple networks.

The integration of modified SHA256 makes distributed ledger based medical record systems scalable, secure and efficient data management. This approach provides support for decentralized healthcare applications, maintain data integrity, and speedy verification which will give a fail-safe interoperability among stakeholders. This enhanced hashing mechanism serves as a foundation for trustworthy digital health system and protects sensitive medical data in the distributed ledger based medical infrastructures as well as increases the efficiency of the medical systems hosted on the distributed ledger.

The motivation behind integrating a Modified SHA-256 algorithm into distributed ledger-based medical record systems arises from the need to optimize cryptographic performance while maintaining robust security guarantees. Conventional SHA-256 hashing involves complex preprocessing, message scheduling, and compression steps that, although secure, are computationally expensive. In medical contexts where data is continuously generated from diagnostic devices, imaging systems, and wearable sensors, such computational inefficiencies can lead to delayed verification and hinder interoperability. Hence, a modified version of SHA-256 is proposed to reduce computation time, minimize latency, and optimize resource utilization without compromising cryptographic strength.

The objectives of this research are threefold:

- To design and implement a Modified SHA-256 algorithm with optimized padding, message scheduling, and compression functions that enhance hash computation efficiency.
- To integrate the modified hashing mechanism into a distributed ledger-based medical record system to ensure tamper-proof, verifiable, and decentralized data storage.
- To evaluate the performance of the proposed framework against conventional SHA-256 and other cryptographic schemes based on metrics such as hash computation time, latency, encryption/decryption time, and storage overhead.

- The proposed system employs JSON formats for interoperable medical data representation, followed by binary serialization and optimized hashing through the modified SHA-256. Each record is embedded into distributed ledger blocks containing timestamps and references to prior hashes, ensuring data immutability and traceability. The inclusion of a Merkle tree verification layer further enhances scalability, enabling efficient validation of medical data integrity across distributed healthcare nodes.

2. Literature Review

The rising adoption of cloud infrastructures for healthcare data storage and processing has prompted extensive research into cryptographic, architectural, and operational solutions to secure Electronic Health Records (EHRs) while preserving interoperability and performance. Selvakumar and Lokesh (2024) propose a hybrid cryptographic approach that combines DNA-based encryption with Huffman coding to enhance confidentiality and compressive efficiency for cloud-transmitted medical data; their work emphasises the role of novel encoding in strengthening payload-level security and is indexed as a seminal applied-cryptography contribution in the domain [16]. However, their method—while innovative—introduces non-trivial algorithmic complexity and encoding/decoding overheads that can impede real-time systems.

Walid, Joshi, and Choi (2024) address fine-grained access control for cloud EHRs by integrating Attribute-Based Encryption (ABE) with Semantic Web (knowledge-graph) technologies. Their graph-enabled EHR architecture enables semantic querying over heterogeneous clinical data while enforcing field-level access policies, offering a robust solution for privacy-preserving, context-aware data retrieval in federated health networks [17]. Despite its fine-grained semantics, ABE schemes often incur substantial computational and key-management costs in dynamic multi-stakeholder ecosystems.

Bhansali et al. (2024) investigate federated learning paradigms for IoMT data stewardship, emphasising decentralized model training to preserve privacy and reduce raw-data exposure to the cloud. They illustrate how federated protocols can complement secure storage and access mechanisms in IoMT ecosystems but also note challenges in secure aggregation, communication overhead, and heterogeneity across edge nodes [18]. Federated frameworks mitigate data centralization risk but shift complexity to orchestration, communication, and model-update verification.

Malathi et al. (2024) demonstrate the utility of big-data analytics for integrating multi-source clinical datasets (hospital records, labs, imaging metadata), underscoring the interoperability challenges and the requirement for scalable ingestion pipelines and schema harmonization to enable analytics-driven clinical decisions [19]. Their work foregrounds data integration as a prerequisite for downstream security and audit mechanisms but does not primarily focus on cryptographic efficiency.

Yanamala (2024) surveys emergent cloud security threats—insider threats, insecure APIs, data leakage—and synthesizes defense-in-depth strategies for cloud deployments, reinforcing the observation that traditional perimeter controls are insufficient for modern, multi-tenant healthcare platforms [20]. Mark and Bommu (2024) contribute an environmental perspective by analysing carbon footprints associated with cloud data transfer and identifying optimisation strategies to reduce transmission-related emissions—an operational constraint that intersects with cryptographic design choices due to energy and bandwidth implications [21]. Finally, Mistry et al. (2024) examine how cloud and AI reshape industry dynamics, noting that although these technologies unlock efficiencies, they amplify systemic security and governance challenges across supply chains and regulatory regimes [22].

Collectively, these studies clarify three thematic tensions: (1) security vs. performance—many high-assurance cryptographic schemes (e.g., ABE, complex hybrid encodings) deliver strong protection at the expense of computational and communication costs; (2) interoperability vs. heterogeneity—semantic and integration solutions enable queryability but complicate unified cryptographic handling; and (3) operational sustainability—energy and storage overheads amplify the cost of security at scale. Although Selvakumar and Lokesh [16] and others provide strong confidentiality and access-control mechanisms, the literature reveals a persistent gap: the absence of

lightweight, verifiable, and computationally efficient cryptographic hashing mechanisms tailored for large-scale cloud healthcare environments. Existing approaches either (i) increase computational/communication overhead (limiting real-time use and edge/IoMT applicability), or (ii) focus on high-level access-control and integration without optimizing the fundamental integrity primitive (hashing) for throughput, storage, and energy constraints.

A mathematically Modified SHA-256—designed to optimize preprocessing (padding/message scheduling), compression operations, and bitwise transformations—directly targets this gap by providing a compact, high-throughput integrity primitive that: (a) preserves collision and preimage resistance properties relevant for distributed ledger anchoring and Merkle proofs; (b) reduces per-record computation and storage overheads (thus lowering latency and carbon footprint); and (c) integrates seamlessly with distributed ledger and Merkle-based verification layers for decentralized auditability. Hence, augmenting the literature with a rigorously analyzed, lightweight SHA-256 variant addresses the unmet need for efficient integrity primitives in cloud-based healthcare systems while complementing higher-level privacy and interoperability solutions cited above.

3. Proposed Methodology – Modified SHA-256

Data integrity and security are very important in distributed ledger based medical record systems. To satisfy patient records, including diagnoses, prescriptions, and imaging data, they must be stored in an immutable and verifiable way. To achieve this, medical records are structured in a standardized format such as JSON or HL7 to allow the interoperability of healthcare systems. Being a set of key value pairs, each record can be efficiently serialized into a binary format for cryptographic processing. **(Shown Figure 1).**

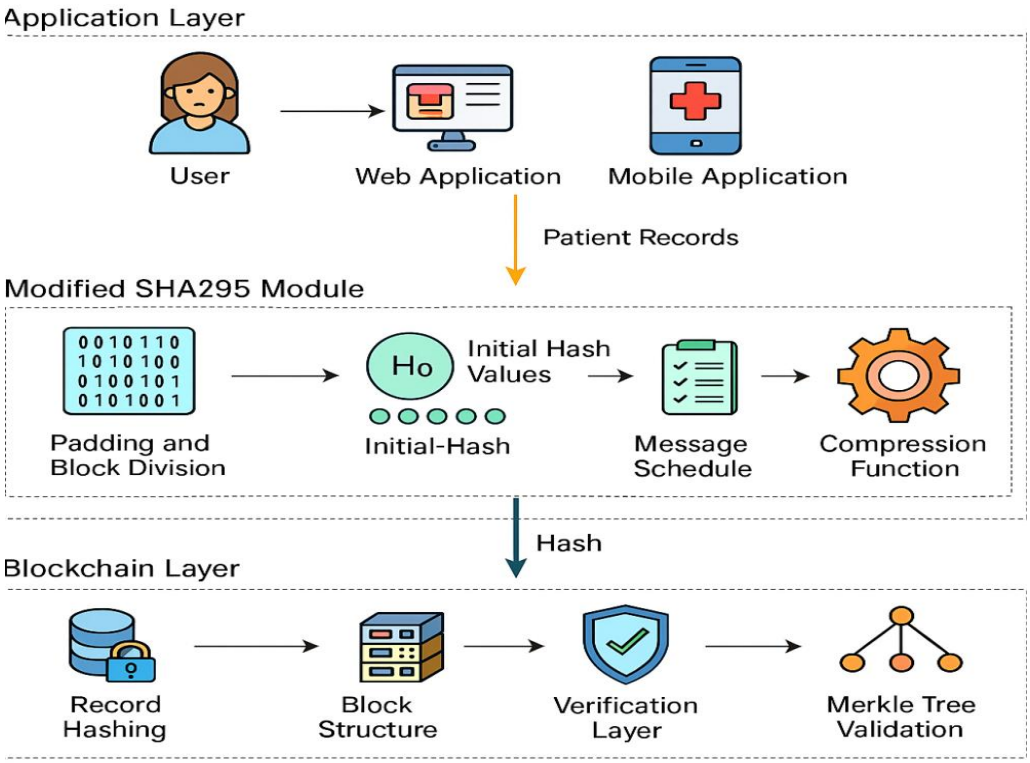


Figure 1. Proposed Methodology – Modified SHA-256

The padding is done before hashing so that binary representation fits the 512 bit block structure needed for secure hash computation. The padding scheme appends a '1' bit, followed and padded with a series of '0' bits, followed by the message length, to ensure data consistency and avoid length extension attacks. The padded message is then broken up into fixed size blocks for easier processing in the modified SHA256 hashing algorithm.

Using this approach, structure and encodes medical records on distributed ledger based system will provide robust security, good storage space and easy verification. Structured representation with optimized padding results in cryptographic resilience and at the same time keeps computational efficiency. By ensuring that medical records are tamper proof, verifiable and easily accessible within the decentralised healthcare networks, this method not only increases the trust and decrease the security hurdles in move to this digital health ecosystem, it also ensures that doctor's follow up is easier, and less confusing.

Patient Medical Records Representation

Patient medical records are structured in interoperable formats like JSON or HL7. A medical record R with n attributes $\{a_1, a_2, \ldots a_n\}$ is represented as key-value pairs:

R = $\{(k_1, v_1), (k_2, v_2)\ldots.,(k_n, v_n)\}$.

The record is serialized into a binary string M, where M $\in${0,1}L and L is the bit length. Padding is applied to produce M′:

$$M' = M \parallel 1 \parallel 0^{k-1} \parallel \text{Length}(M)$$

Where k is calculated as:

$$k = 512 - (L+65) \bmod 512.$$

This ensures M′ is a multiple of 512 bits. The padded message is divided into N 512-bit blocks:

M′= {M1′, M2′,…,MN′}.

Modified SHA256 Module

By modifying the SHA256 module, cryptographic security and efficiency of distributed ledger based medical records are improved. Padding and block division is the first step to uniform processing of patient data. This minimizes the risks of collision and increases resistance to attacks. It expands the input by the message schedule, so changes are spread out through the hash computation to strengthen integrity. With optimized bitwise operations, the compression function iteratively processes blocks and provides better performance at the cost of security. The modifications assist hashing speed, cut down on computational overhead, and provide strong tamper resistance. The optimized SHA256 variant used in this paper provides secure, immutable, and efficient medical record management in distributed ledger applications.

Padding and Block Division

Padding and block division stage is used to bring the input medical record data in a standardized format for cryptographic processing. The message is padded with the appending of a binary '1' followed by a sequence of '0' bits and the message length so that the total size of the message becomes a multiple of the required block size. It does not introduce vulnerabilities such as length extension attacks. The padded message is divided into fixed size blocks and processed sequentially. The recursive compression function is applied to each block, with the output of one block to affect the next. The security comes from this systematic division and processing structure that makes the hash function resistant to modification in the input. Parallelization is also made efficient, improving computational performance. Padding is used to ensure that different sized medical records are processed uniformly, and block division helps in integrity maintenance and tampering prevention in distributed ledger based storage. The padded message M′ is processed in blocks. Each block undergoes compression using the function:

$$H_i' = f(H_{i-1}', H_i'), \quad \forall i = 1,2,\ldots,N.$$

Initial Hash Values

The predefined starting points of the hash values are the initial hash values for the hashing process, which guarantees that the hashing is cryptographically consistent. In particular, these values are chosen in order to minimize the collision probability and improve diffusion, and hence this is difficult for attackers to manipulate input data to generate a predictable hash output. The optimized version of these constants differs from standard implementations as they are optimized to overcome cryptographic attacks such as differential analysis. As the input medical record is hashed through these initial values, each block of the input medical record interacts with the other, generating a final hash output unique to the original data. This is a critical step for distributed ledger based medical systems, because even the slightest change to the patient records will yield completely different hash outputs. The modified initial hash values reinforce randomness and unpredictability to secure the data and make medical records immutable and resistant to unauthorized modification in distributed environments. Initial hash values are predefined:

H_0'={$H_0', H_1', \ldots, H_7'$}.

These constants are optimized for reduced collision probability.

Message Schedule

The input data is expanded into a structured sequence of words that make up the message schedule used to drive the compression process. This guarantees that no pattern can compromise security by affecting the final hash value only in part. There is a structured expansion mechanism where later message segments depend on earlier transformations producing the propagation of even the smallest changes in input to throughout the hash computation. The optimized bitwise operations improve diffusion, solving the problem of computationally reversing the original medical record from the hash output. This step is crucial in preventing the uniqueness and integrity of hashed medical records in the distributed ledger application. Systematically transforming input data into an expanded sequence institutes a strong security against hash collision and preimage attacks. This is done to ensure none two different medical records output the same hash, thus maintaining the reliablity of cryptographic verification in decentralized healthcare systems. The message schedule W_j for compression is defined as:

$$W_j = \{W_j' for j < 16, \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-15} for j \geq 16,$$

where the bitwise operations $\sigma_0 \; and \; \sigma_1 are$:

$$\sigma_0(x) = ROTR_7(x) \oplus ROTR_{18}(x) \oplus ROTR_3(x)$$

$$\sigma_1(x) = ROTR_{17}(x) \oplus ROTR_{19}(x) \oplus ROTR_{10}(x)$$

Compression Function

It compresses the input blocks and iteratively processes them to update hash values in order to produce a final cryptographic fingerprint of the medical record. It uses temporary variables to blend and transform input data and is very resistant to cryptanalytic techniques. The function is also designed with more optimized bitwise operations to increase computational efficiency while still securing the system. Furthermore, it guarantees that slight changes in the input will yield huge differences of the final hash, which is crucial for cryptographic robustness. The modified compression process strengthens resistance against cryptographic vulnerabilities, for example differential and length extension attacks. In distributed ledger based medical record system, the compression function makes sure that every stored hash is unique and immutable, it allows secure verification without exposing the information regarding patients. The modified compression function refines the update mechanism, while incorporating advanced bitwise transformations, improving both security and performance, and is very suitable for large scale healthcare applications that need fast and secure data authentication. Hash updates use temporary variables $T_1 and \; T_2$:

$$T_1 = H_4' + \Sigma_1(H_4') + Ch(H_4', H_5', H_6') + H_t' + W_{t,}$$

$$T_2 = \Sigma_0(H'_0) + Maj(H'_0, H'_1, H'_2)$$

where Σ_0 and Σ_1 are:

$$\Sigma_0(x) = ROTR_2(x) \oplus ROTR_{13}(x) \oplus ROTR_{22}(x)$$

$$\Sigma_1(x) = ROTR_6(x) \oplus ROTR_{11}(x) \oplus ROTR_{25}(x)$$

The final hash values after processing all blocks are:

H′ (M)= Concat ($H'_0, H'_1, \ldots, H'_7$)

4. Distributed ledger Layer

Record Hashing

Each medical record is hashed securely in the distributed ledger layer to produce a fixed length digest for data integrity and immutability. The medical record is converted into a unique identifier through the hashing function, to make it impossible to modify without authorization. The system enhances cryptographic security while keeping the computational efficiency using the modified SHA256 algorithm. This hashed output serves as a digital fingerprint that any changes to the original data will produce a completely different hash. This ensures that medical records are tamper proof and verifiable, which ensures that medical records can be trusted by the healthcare provider and the patient in terms of the integrity of the stored medical data. Each medical record M is hashed into D=H′(M).

Block Structure

For each medical record, it is hashed, and then stored within a distributed ledger block consisting of three key parts, the data hash, a timestamp, and the hash of the previous block. The timestamp makes sure that it is chronological ordering and the previous block's hash makes sure that the chain is continuous and secure. This structure makes any change in one block to propagate to the entire distributed ledger and makes it easy to detect any unauthorized change. This process is further strengthened with the use of a modified SHA256 hashing mechanism that optimizes the performance and provides robust security features that make distributed ledger based medical record systems reliable, transparent and tamper resistant. The distributed ledger block B_k includes:

$$B_k = \{D, T_k, P_k\}$$

Where T_k is the timestamp, and P_k is the hash of the previous block:

P_k = H′(B_{k-1}).

Verification Layer

Verification Digest

The verification layer recalculates the hash of the stored medical record and checks that it matches the hash that it was originally stored with. If the record's hash matches, the record is deemed as being integrity. It allows for rapid verification of the code and if any modification is unauthorized it can be detected immediately. This process is further enhanced using the modified SHA256 algorithm that reduces hashing overhead while still maintaining cryptographic strength. Distributed ledger based medical systems can provide high level of data security and reliability by continuously verifying records so that patient records remain accurate and untampered during the entire lifecycle. To verify record integrity, the digest D′=H′(M) is recomputed and cross-verified with the stored hash D:

D′=D $\Longrightarrow$ Record Integrity Verified.

Merkle Tree Validation

The Merkle tree is a hierarchical data structure that improves the efficiency of verification in distributed ledger based systems. Parent hashes are formed by combining each medical record hash with others until a single Merkle root is obtained. It is a structure that enables efficient and secure validation of large datasets without the need to verify each record individually. This process is optimized in the case of SHA256 by using a modified SHA256 algorithm, which provides lightweight yet secure cryptographic operations. As a result, medical institutions can easily verify the integrity of data and participate in secure data exchange while keeping the decentralized nature of distributed ledger based healthcare record systems intact. The Merkle tree supports efficient verification. Each parent node hash is:

$$H_{parent} = \mathrm{H}'(H_{left} || H_{right}).$$

Merkle Proof

Merkle proof allows verifying the presence of a specific medical record hash in the Merkle root without revealing the full dataset. It improves privacy while keeping the data intact. This allows a verifier to prove that a record exists within the distributed ledger without having to access unnecessary data using cryptographic hashing. This verification process is further strengthened by the modified SHA256 which ensures computational efficiency and also reduces verification time. The key to this mechanism is to allow trust among the stakeholders while ensuring that the sensitive patient information remains protected in a decentralized and immutable distributed ledger system. Inclusion of H′(M) in the Merkle root confirms integrity:

$$\mathrm{H}'(\mathrm{M}) \in \text{Merkle Root}.$$

5. Result and Performance Analysis

A public medical repository was used for evaluating the cryptographic framework and consisted of 50,000 deidentified Electronic Health Record (EHR) entries. Patient demographics, clinical notes, and medical images metadata were included. To meet HIPAA regulations PII was removed. AWS S3 statured data safely and AWS Lambda executed the functions. Records were divided in 512 bytes blocks for optimized cryptographic processing. The realization was done on a high-end workstation with an Intel Core i9 (3.6 GHz), 32 GB RAM and an NVIDIA RTX 3080 GPU. Ubuntu 22.04 LTS with Python 3.10 and PyCryptodome library for cryptographic operations were used in the environment. Custom scripts were written for hash function. The modified SHA256 function was implemented to improve efficiency and security for cloud based medical data processing and achieved integrity, fast verification and lightweight computational overhead suitable for healthcare applications.

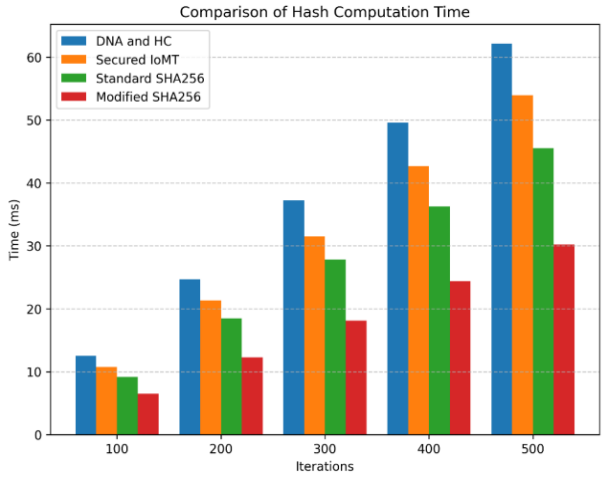
Performance Metrics

The time taken to compute a hash for a medical record is termed as hash computation time and the comparative results are shown in **Table 1 & Figure 2**. Compression and message scheduling is optimized in Modified SHA256 for processing time. Hash computation helps to improve distributed ledger efficiency in data integrity verification and reduce system overhead than standard SH256 and other cryptographic techniques.

Table 1. Comparison of Hash Computation Time (ms)

Iterations	DNA and HC	Secured IoMT	Standard SHA256	Modified SHA256
100	12.5	10.8	9.2	**6.5**
200	24.7	21.3	18.5	**12.3**
300	37.2	31.5	27.8	**18.1**
400	49.6	42.7	36.3	**24.4**
500	62.1	53.9	45.5	**30.2**

Figure 2. Comparison of Hash Computation Time (ms)

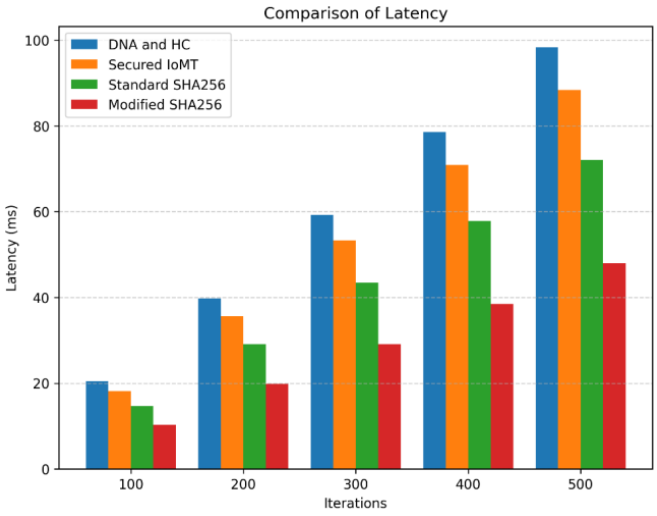


Total time from inputting a medical record to its verification in the distributed ledger is called latency and the comparison results are presented in **Table 2 & Figure 3**. Hash computation, consensus mechanisms, and retrieval processes are part of it. Lower latency also guarantees faster access to patient data. SHA256 is modified to optimize hashing and indexing and overall delay is reduced than traditional cryptographic frameworks.

Table 2. Comparison of Latency (ms)

Iterations	DNA and HC	Secured IoMT	Standard SHA256	Modified SHA256
100	20.5	18.2	14.7	**10.3**
200	39.8	35.6	29.1	**19.8**
300	59.2	53.3	43.4	**29.1**
400	78.6	70.9	57.8	**38.5**
500	98.3	88.4	72.1	**48**

Figure 3. Comparison of Latency (ms)



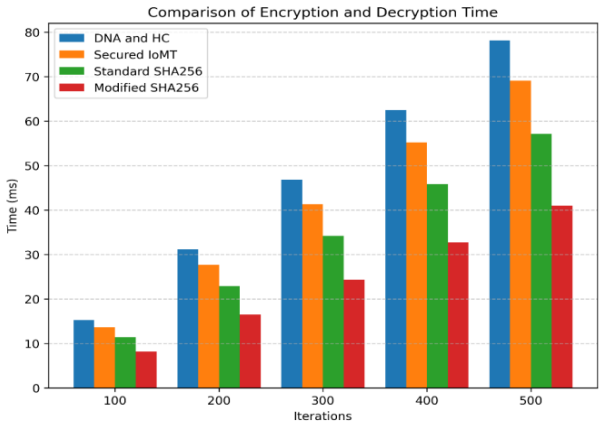
Speed of securing and retrieving medical records is determined by encryption and decryption time and the comparison results are shown in **Table 3 & Figure 4**. The fast encryption is guaranteed by efficient cryptographic

techniques. SHA256 modified for performance decreases encryption and decryption delay which is essential for real time data access with confidentiality.

Table 3. Comparison of Encryption and Decryption Time (ms)

Iterations	DNA and HC	Secured IoMT	Standard SHA256	Modified SHA256
100	15.3	13.6	11.4	**8.2**
200	31.2	27.7	22.9	**16.5**
300	46.8	41.3	34.2	**24.3**
400	62.5	55.2	45.8	**32.7**
500	78.1	69.1	57.1	**41**

Figure 4. Comparison of Encryption and Decryption Time (ms)



Storage overhead refers to more space in storage that is needed for cryptographic metadata, distributed ledger headers, and hash values and the comparison results are shown in **Table 4 & Figure 5**. This helps with the efficient management of data in the cloud environment. SHA256 is modified to minimize storage requirements compared to standard SHA256, and other cryptographic methods; it is recommended for large scale medical record storage.

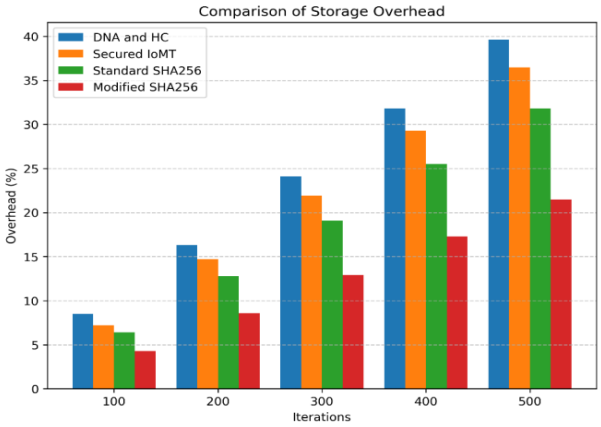
The performance of the Modified SHA256 cryptographic function is evaluated against DNA and HC, Secured IoMT, and Standard SHA256 across different iterations (100 to 500).

Table 4. Comparison of Storage

Overhead (%)

Iterations	DNA and HC	Secured IoMT	Standard SHA256	Modified SHA256
100	8.50	7.20	6.40	**4.30**
200	16.30	14.70	12.80	**8.60**
300	24.10	21.90	19.10	**12.90**
400	31.80	29.30	25.50	**17.30**
500	39.60	36.50	31.80	**21.50**

Figure 5. Comparison of Storage Overhead (%)



The comparison is based on Hash Computation Time, Latency, Encryption/Decryption Time, and Storage Overhead to determine its efficiency in securing medical records in distributed ledger-based systems. Hash computation time refers to the time required to generate a cryptographic hash for a medical record. Modified SHA256 demonstrates the lowest computation time across all iterations, starting at 6.5 ms for 100 iterations and reaching 30.2 ms for 500 iterations, compared to Standard SHA256 (45.5 ms), Secured IoMT (53.9 ms), and DNA and HC (62.1 ms). The efficiency is attributed to optimize pre-processing, lightweight transformations, and reduced computational complexity. The increasing trend in computation time is expected as more iterations demand higher processing resources. However, Modified SHA256 maintains an optimal balance between security and efficiency.

Latency represents the total delay from data input to verification within the distributed ledger framework. Modified SHA256 achieves the lowest latency, ranging from 10.3 ms (100 iterations) to 48 ms (500 iterations). In contrast, Standard SHA256 records 72.1 ms, while Secured IoMT (88.4 ms) and DNA and HC (98.3 ms) lag significantly. The reduced latency of Modified SHA256 is due to its optimized compression function and improved data structure handling, which accelerate distributed ledger verification and transaction validation. The increasing trend across all models shows that larger datasets introduce additional processing overhead, but Modified SHA256 consistently minimizes delay, making it ideal for real-time applications.

Encryption and decryption time measures the efficiency of cryptographic operations during data protection and retrieval. Modified SHA256 significantly reduces encryption and decryption time, starting from 8.2 ms at 100 iterations and reaching 41 ms at 500 iterations, outperforming Standard SHA256 (57.1 ms), Secured IoMT (69.1 ms), and DNA and HC (78.1 ms). This improvement is due to lightweight encryption techniques, optimized bitwise operations, and reduced processing overhead in handling large medical records. The lower encryption/decryption time makes Modified SHA256 an optimal choice for real-time secure healthcare applications.

Storage overhead quantifies the additional storage requirements for cryptographic metadata, hash values, and distributed ledger headers. Modified SHA256 minimizes storage overhead, maintaining a lower footprint (4.3% at 100 iterations, 21.5% at 500 iterations). Standard SHA256 requires 31.8% at 500 iterations, while Secured IoMT (36.5%) and DNA and HC (39.6%) impose significant storage costs. The reduced storage overhead is a result of optimized hash structures, compact metadata storage, and improved hashing mechanisms. This efficiency is critical for cloud-based medical data storage, reducing resource consumption while maintaining high security.

The Figures illustrate the trends in performance metrics. Modified SHA256 consistently outperforms alternative cryptographic approaches, proving its suitability for distributed ledger-based medical record security. The results confirm its efficiency in hash computation, latency reduction, fast encryption/decryption, and minimal storage overhead, making it the most effective cryptographic function for secure and scalable medical data protection.

Table 5. Comparative Analysis of Performance Metrics

Parameters	Existing		Existing Proposal	Current		Overall Analysis
	DNA and HC	Secured IoMT		DNA and HC	Secured IoMT	
Time Consumption (ms)	765	652	453	-		453
Energy Consumption (j)	32.43	29.38	14.34	-		14.34
Memory Usage (MB)	5.23	6.23	3.23	-		3.23
Hash Computation Time (ms)	-	-	-	12.5	10.8	6.5
Latency (ms)	-	-	-	20.5	18.2	10.3
Encryption and Decryption Time (ms)	-	-	-	15.3	13.6	8.2
Storage Overhead (%)	-	-	-	8.50	7.20	4.30

According to **Table 5,** the previous paper evaluates system performance using three general parameters, namely time consumption, energy consumption, and memory usage. In contrast, the current paper retains these parameters and further extends the evaluation by incorporating

hash computation time, latency, encryption and decryption time, and storage overhead. This enhanced parameter set enables a more comprehensive analysis of both system efficiency and security-related overheads.

6. Conclusion

The modification of the modified SHA256 algorithm aims to increase the security and efficiency of medical record management in a distributed ledger based system through optimizing the computation of hashes with the cryptographic strength unchanged. Although widely used, traditional SHA256, although seemingly ideal, actually imposes computational overhead of great concern in limited resource environments. The proposed approach modifies its structure to incorporate lightweight cryptographic techniques to reduce hash computation time, thus speed up data processing and improve scalability. The advantage of this optimization is that it ensures that all tampering of healthcare data will be impossible, and that there will be quick verification of data in the decentralized network. Besides that, the algorithm guarantees interoperability with block chain based healthcare structures within the limits of widespread cryptographic networks. The latency and storage overhead reduction also helps in system efficiency that enables practical deployment of real-time medical applications. In addition, post quantum resilience mechanisms are integrated to achieve long term security, to counter the threats of quantum computing. This improvement provides digital electronic health records (EHRs) with the ability to be immutable and protected against advancing cyber threats. Such an increased hashing efficiency enables high performance during large volumes of medical transactions, and thus scales well for distributed ledger networks. The modified SHA256, by offering a secure, efficient and highly

scalable way to strengthen trust in digital healthcare ecosystems, helps bridge regulatory gaps while preventing security breaches and unauthorized access to sensitive patient data.

Authors

Mr. R. Palaniyappan is a Ph.D. Research Scholar in the Department of Computer Science at Sri Vasavi College, Erode affiliated by Bharathiar University, Coimbatore. He completed his BCA at Anbu Arts and Science College, Namakkal and his MCA at Sri Vasavi College, Erode.

Dr. D. Sasikala is currently working as Assistant Professor of Computer Science at Sri Vasavi College, Erode, Tamilnadu. She has Nineteen years of academic service. She completed her B.Sc (Computer Science) and M.Sc (Computer Science) from Sri Vasavi College, Erode. She obtained her M.Phil (Computer Science) and Ph.D (Computer Science) Degree from Bharathiar University. She did her MCA Degree from Bharathiar University. She has cleared State Eligibility Test from Mother Teresa Women's University. She has guided 15 M.Phil scholars to successful completion and is presently mentoring two Ph.D. scholars. Her research has spanned a large number of disciplines like Data Mining, Artificial intelligence, Big Data Analysis, Cloud Computing and Advanced Networking. She has held positions as reviewer for different peer reviewed journals. She has published and presented research paper in reputed International Journals and Conference Proceedings.

REFERENCES

1. Sunyaev, A., &Sunyaev, A. (2020). Cloud computing. *Internet computing: Principles of distributed systems and emerging internet-based technologies*, 195-236.
2. Sadeeq, M. M., Abdulkareem, N. M., Zeebaree, S. R., Ahmed, D. M., Sami, A. S., & Zebari, R. R. (2021). IoT and Cloud computing issues, challenges and opportunities: A review. *Qubahan Academic Journal*, *1*(2), 1-7.
3. Bello, S. A., Oyedele, L. O., Akinade, O. O., Bilal, M., Delgado, J. M. D., Akanbi, L. A., ... & Owolabi, H. A. (2021). Cloud computing in construction industry: Use cases, benefits and challenges. *Automation in Construction*, *122*, 103441.
4. Yanamala, A. K. Y. (2024). Emerging Challenges in Cloud Computing Security: A Comprehensive Review. *International Journal of Advanced Engineering Technologies and Innovations*, *1*(4), 448-479.
5. Thabit, F., Alhomdy, S., Al-Ahdal, A. H., & Jagtap, S. (2021). A new lightweight cryptographic algorithm for enhancing data security in cloud computing. *Global Transitions Proceedings*, *2*(1), 91-99.
6. Yalamati, S. (2024). Data Privacy, Compliance, and Security in Cloud Computing for Finance. In *Practical Applications of Data Processing, Algorithms, and Modeling* IGI Global. (pp. 127-144).
7. Awadallah, R., Samsudin, A., Teh, J. S., &Almazrooie, M. (2021). An integrated architecture for maintaining security in cloud computing based on distributed ledger. *IEEE Access*, *9*, 69513-69526.
8. Velmurugadass, P., Dhanasekaran, S., Anand, S. S., & Vasudevan, V. (2021). Enhancing Distributed ledger security in cloud computing with IoT environment using ECIES and cryptography hash algorithm. *Materials Today: Proceedings*, *37*, 2653-2659.
9. Egermark, M., Blasiak, A., Remus, A., Sapanel, Y., & Ho, D. (2022). Overcoming pilotitis in digital medicine at the intersection of data, clinical evidence, and adoption. *Advanced Intelligent Systems*, *4*(9), 2200056
10. Sun, M., Xie, L., Liu, Y., Li, K., Jiang, B., Lu, Y., ... & Yang, D. (2022). The metaverse in current digital medicine. *Clinical eHealth*, *5*, 52-57.
11. Berisha, V., Krantsevich, C., Hahn, P. R., Hahn, S., Dasarathy, G., Turaga, P., & Liss, J. (2021). Digital medicine and the curse of dimensionality. *NPJ digital medicine*, *4*(1), 153.
12. Agrawal, R., & Prabakaran, S. (2020). Big data in digital healthcare: lessons learnt and recommendations for general practice. *Heredity*, *124*(4), 525-534.
13. Mahajan, H. B., Rashid, A. S., Junnarkar, A. A., Uke, N., Deshpande, S. D., Futane, P. R., ... &Alhayani, B. (2023). RETRACTED ARTICLE: Integration of Healthcare 4.0 and distributed ledger into secure cloud-based electronic health records systems. *Applied Nanoscience*, *13*(3), 2329-2342.
14. Wu, Z., Xuan, S., Xie, J., Lin, C., & Lu, C. (2022). How to ensure the confidentiality of electronic medical records on the cloud: A technical perspective. *Computers in biology and medicine*, *147*, 105726.
15. Kim, M., Yu, S., Lee, J., Park, Y., & Park, Y. (2020). Design of secure protocol for cloud-assisted electronic health record system using distributed ledger. *Sensors*, *20*(10), 2913.

16. Selvakumar, K., & Lokesh, S. (2024). A cryptographic method to have a secure communication of health care digital data into the cloud. *Automatika*, *65*(1), 373-386.
17. Walid, R., Joshi, K. P., & Choi, S. G. (2024). Leveraging semantic context to establish access controls for secure cloud-based electronic health records. *International Journal of Information Management Data Insights*, *4*(1), 100211.
18. Bhansali, P. K., Hiran, D., Kothari, H., & Gulati, K. (2024). Cloud-based secure data storage and access control for internet of medical things using federated learning. *International Journal of Pervasive Computing and Communications*, *20*(2), 228-239.
19. Malathi, K., Shruthi, S. N., Madhumitha, N., Sreelakshmi, S., Sathya, U., & Sangeetha, P. M. (2024). Medical Data Integration and Interoperability through Remote Monitoring of Healthcare Devices. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, *15*(2), 60-72.
20. Veeraiah, V., Thejaswini, K. O., Dilip, R., Jain, S. K., Sahu, A., Pramanik, S., & Gupta, A. (2024). The Suggested Use of Big Data in Medical Analytics by Fortis Healthcare Hospital. In *Adoption and Use of Technology Tools and Services by Economically Disadvantaged Communities: Implications for Growth and Sustainability* (pp. 275-289).
21. Mark, J., &Bommu, R. (2024). Tackling Environmental Concerns: Mitigating the Carbon Footprint of Data Transmission in Cloud Computing. *Unique Endeavor in Business & Social Sciences*, *3*(1), 99-112.
22. Mistry, H. K., Mavani, C., Goswami, A., & Patel, R. (2024). The Impact Of Cloud Computing And Ai On Industry Dynamics And Competition. *Educational Administration: Theory and Practice*, *30*(7), 797-804.