A Robust Security Framework for Cloud Data Storage Using Lightweight Blockchain Technology

Renuka GOLLA BALA[1*], S.Gnanavel[2]
Department of Computing Technologies, SRM Institute of Science and Technology,
Kattankulathur, Chennai, Tamil Nadu - 603203, India

Abstract—The exponential growth of cloud computing has enabled large-scale data outsourcing but has simultaneously introduced critical challenges related to data confidentiality, integrity, and trust. Traditional cryptographic and blockchain-based cloud security solutions often suffer from high computational overhead, latency, and scalability limitations, which hinder their practical adoption. To address these issues, this study proposes a robust and lightweight blockchain-based security framework for secure cloud data storage. The framework integrates hybrid AES–ECC encryption, smart contract–driven access control, and a lightweight consensus mechanism combining Delegated Proof of Stake (DPoS) and Practical Byzantine Fault Tolerance (PBFT) to achieve efficient and tamper-resistant data management. The proposed system employs an on-chain/off-chain hybrid architecture that stores only essential metadata and cryptographic proofs on the blockchain while maintaining the actual data in distributed cloud storage. This design minimizes computational burden and blockchain bloat while ensuring end-to-end transparency and verifiability. A Merkle tree–based Proof of Storage (PoS) mechanism enables rapid integrity verification without requiring full data retrieval. Comprehensive experiments were conducted using a simulated multi-node cloud environment to evaluate encryption efficiency, transaction latency, throughput, storage overhead, and energy consumption. Results show that the proposed framework outperforms existing blockchain-based models, achieving a 37.7% reduction in encryption/decryption time, a 51.3% decrease in transaction latency, and a 54.5% improvement in energy efficiency. Additionally, the system attained a 99.3% security success rate under various attack scenarios, demonstrating its resilience against unauthorized access, replay, and tampering attempts. These findings confirm that the proposed approach provides a practical balance between security assurance and performance optimization.

Keywords—Cloud data security; lightweight blockchain; AES–ECC encryption; smart contracts; Proof of Storage (PoS); data integrity verification; energy-efficient consensus

I. INTRODUCTION

The rapid expansion of cloud computing services has enabled organizations and individuals to outsource large volumes of data to third-party storage platforms, yielding dramatic gains in scalability and operational flexibility. At the same time, reliance on remote storage raises persistent concerns about confidentiality, integrity, and trustworthy access control, particularly when storage providers are not fully trusted [10, 2]. Traditional cryptographic techniques while effective for confidentiality at the data level, do not by themselves provide an auditable, tamper-resistant record of storage operations, nor do they always scale gracefully when integrity verification, access logging, and multi-party accountability are required [5, 2].

Blockchain technology has emerged as a promising substrate to fill this gap by providing an immutable ledger for transaction logging, access policy enforcement, and distributed trust anchoring; several surveys have documented the potential of combining blockchain and cloud storage for stronger auditability and decentralized authorization [8,11]. Nevertheless, mainstream blockchain designs often impose prohibitive computational and storage costs, especially when classical Proof-of-Work consensus is used or when full transaction histories are stored on every node, rendering direct adoption impractical for latency-sensitive cloud services or resource-constrained edge/cloud hybrid deployments [3,6]. Consequently, achieving the security benefits of blockchain in cloud settings requires careful design choices that minimize on-chain state and lower consensus overhead without compromising cryptographic guarantees [9,7].

A second important class of techniques addresses data possession and retrievability with compact protocols that allow a verifier to check that an untrusted storage provider retains data intact without downloading the entire file. Provable Data Possession (PDP) and Proofs of Retrievability (POR) form this foundation: PDP schemes enable probabilistic integrity checks using small challenge–response protocols, while PORs augment these ideas with mechanisms to guarantee full recoverability under bounded corruption [1,2,10,5]. Integrating such PoS (proof-of-storage) building blocks with an immutable blockchain ledger yields a compelling design in which integrity proofs and their verification outcomes are anchored on-chain while bulky data remains off-chain—thus avoiding ledger bloat while preserving verifiability [16].

For confidentiality and key management, hybrid cryptographic constructions that combine fast symmetric ciphers (e.g., AES) for bulk encryption with elliptic-curve cryptography (ECC) for secure, compact key encapsulation provide attractive tradeoffs between performance and security. Empirical studies and applied proposals have shown that AES–ECC hybrids deliver strong confidentiality with modest computational cost, which is particularly valuable in cloud settings where large volumes of data are encrypted frequently [12, 14]. At the same time, the use of elliptic-curve signatures and key exchange supports digital authentication and non-repudiation for access requests recorded via smart contracts [6, 3].

*Corresponding author.

A key systems design principle emerging from recent literature is the on-chain/off-chain hybrid architecture: only succinct metadata, Merkle-root hashes, and verification proofs are stored on the blockchain, while the data itself is retained in conventional cloud storage or distributed object stores. Merkle trees enable compact integrity references and logarithmic verification cost, and the root hash binds the off-chain content to an immutable on-chain anchor [4]. This hybrid approach reduces ledger growth and enables lightweight nodes to participate in integrity verification without maintaining full data replicas, an important consideration for scalability and energy efficiency [13, 9].

Despite these promising directions, two practical tensions remain. First, consensus mechanisms must be chosen to strike a balance between fault tolerance and resource consumption: permissioned or delegated schemes such as DPoS combined with PBFT-style agreement can deliver low latency and high throughput for permissioned cloud settings, but their resilience properties and selection/rotation policies need rigorous design to mitigate collusion and sustain liveness under churn [6,7]. Second, the integration of smart contracts for fine-grained access control introduces new attack surfaces and operational challenges that require conservative contract design and formal policy encoding to avoid authorization errors and replay vulnerabilities [15].

Despite substantial advances in blockchain-enabled cloud security, existing frameworks continue to face significant challenges related to scalability, consensus overhead, storage expansion, and energy consumption. Many existing systems rely on computationally intensive consensus mechanisms or maintain excessive on-chain data, resulting in increased transaction latency and reduced practical deployability. Furthermore, achieving a balanced integration of confidentiality, integrity verification, decentralized access control, and auditability remains a persistent challenge in large-scale cloud environments. These limitations motivate the development of lightweight blockchain architectures capable of providing strong security guarantees while maintaining operational efficiency, scalability, and cost-effectiveness [9, 13, 7].

In this study, we synthesize these strands, namely PDP/POR-style proofs, Merkle-anchored on-chain metadata, hybrid AES–ECC encryption, and lightweight delegated consensus into a single coherent framework that aims to preserve the security advantages of blockchain while remaining practical for real cloud deployments. Our contributions are threefold: 1) a modular architecture that minimizes on-chain state by storing only cryptographic digests and PoS outcomes; 2) a hybrid cryptographic and key-encapsulation protocol optimized for cloud throughput; and 3) an evaluation demonstrating that a lightweight blockchain design can deliver strong security guarantees with markedly lower latency and energy cost than traditional blockchain-first approaches. The following sections detail the design rationale, security analysis, implementation choices, and experimental evaluation that collectively validate the framework's efficacy in real-world cloud scenarios.

II. Literature Review

Cloud storage security has been extensively explored over the past decade, particularly regarding data integrity verification and access control in untrusted environments. Early research emphasized third-party auditing and homomorphic authenticators to verify stored data without full retrieval. For instance, Wang et al. [33] introduced a privacy-preserving public auditing mechanism using bilinear aggregate signatures to enable batch verification. These foundational works paved the way for integrating distributed trust into cloud storage ecosystems.

The integration of blockchain into cloud data management represents a major paradigm shift from centralized to distributed trust models. Chen et al. [20] developed a blockchain-assisted auditing framework for shared cloud data that maintains data traceability while minimizing computation cost through Merkle hash optimization. Such frameworks demonstrate the utility of blockchain in creating immutable verification logs for outsourced data.

Lightweight blockchain architectures have been increasingly emphasized to reduce overhead in security-sensitive but resource-constrained environments. For example, Dorri et al. [21] designed a lightweight blockchain protocol tailored for the Internet of Things (IoT), which minimizes storage and energy consumption through hierarchical trust delegation. Xie et al. [36] advanced this concept by proposing a sharded lightweight blockchain for scalable distributed storage, achieving high throughput with minimal block propagation delay. These approaches illustrate the critical balance between security strength and system efficiency that motivates current lightweight blockchain frameworks for cloud environments.

Several studies have examined hybrid blockchain-cloud models that separate data storage from metadata management. Liu et al. [27] proposed a hybrid system in which encrypted cloud data is stored off-chain, while blockchain records only integrity metadata and access logs. Similarly, Wang et al. [34] utilized blockchain smart contracts to manage encrypted cloud metadata dynamically, reducing redundancy and ensuring verifiable deletion. These hybrid architectures serve as a precursor to the on-chain/off-chain hybrid designs employed in modern blockchain-enabled cloud frameworks.

Consensus mechanism optimization remains central to blockchain-based cloud security research. Proof-of-Work (PoW) and Proof-of-Stake (PoS) protocols, though secure, incur significant energy costs and latency [25]. To mitigate these drawbacks, Nguyen et al. [30] proposed a Delegated Proof-of-Stake (DPoS) model for consortium clouds, achieving higher transaction throughput with lower energy consumption. Likewise, Fan et al. [22] employed a Practical Byzantine Fault Tolerance (PBFT) variant in permissioned blockchain networks to improve consensus reliability under malicious fault conditions. These approaches collectively highlight ongoing efforts to develop energy-efficient, low-latency consensus schemes suitable for large-scale cloud infrastructures.

Hybrid cryptographic methods have been proposed to strengthen confidentiality and key management efficiency in

distributed environments. Gao et al. [23] combined ECC and RSA with blockchain authentication for secure key exchange in cloud storage, while Alharbi et al. [18] developed an AES–RSA hybrid model integrated with a blockchain ledger to secure medical records. Studies such as Rahman et al. [31] further emphasize the value of elliptic curve cryptography (ECC) for lightweight yet secure key generation, aligning with the hybrid encryption direction in the present research.

From a data integrity standpoint, lightweight Merkle-tree–based verification remains a crucial building block. Li et al. [26] proposed a dynamic Merkle tree approach to enable partial verification of large files, significantly reducing computational complexity. Similarly, Singh et al. [32] incorporated authenticated skip lists for hierarchical integrity verification, improving scalability and auditability. These studies demonstrate that hierarchical hash structures can efficiently support data integrity auditing when integrated with blockchain proofs.

Smart contract–driven access control is another vital direction. Maesa et al. [29] demonstrated that smart contracts can implement transparent and verifiable access policies across distributed storage systems. Building upon this, Al Omar et al. [19] introduced MediBchain, a blockchain-based framework using smart contracts for secure healthcare data sharing. More recently, Wu et al. [35] presented a policy-based smart contract model to dynamically regulate user access privileges in federated cloud systems. These advances inform the policy automation aspect of modern blockchain-enabled cloud security frameworks.

Energy efficiency and scalability are increasingly prioritized in blockchain security design. Ma et al. [28] optimized node communication overhead using differential consensus weights. Such works demonstrate that lightweight blockchain designs can maintain security while reducing energy and time complexity, a motivation underlying the proposed framework's design philosophy.

Finally, integrated frameworks combining cryptographic verification, blockchain-based consensus, and cloud resource management have emerged as the next frontier. Al-Bassam [17] presented a decentralized trust framework leveraging blockchain for cloud identity management, while Huang et al. [24] proposed a blockchain-assisted privacy-preserving model that supports auditability in multi-cloud systems. These efforts collectively demonstrate the necessity of modular, lightweight, and verifiable architectures, principles embodied in the proposed security framework that aims to unify efficiency, scalability, and trust assurance for cloud data storage.

III. Proposed Method

The proposed method introduces a robust and lightweight blockchain-based framework that ensures the confidentiality, integrity, and authenticity of data stored in cloud environments. The system design aims to overcome traditional limitations of blockchain–cloud integrations, such as computational overhead, latency, and scalability issues, while maintaining strong cryptographic guarantees. The framework operates through the interaction of four integral components: the Data Owner Module, Cloud Service Provider (CSP) Module, Blockchain Security Layer, and User Access Module, which collectively establish a secure, decentralized, and auditable data management process. The architecture of the proposed method is depicted in Fig. 1.

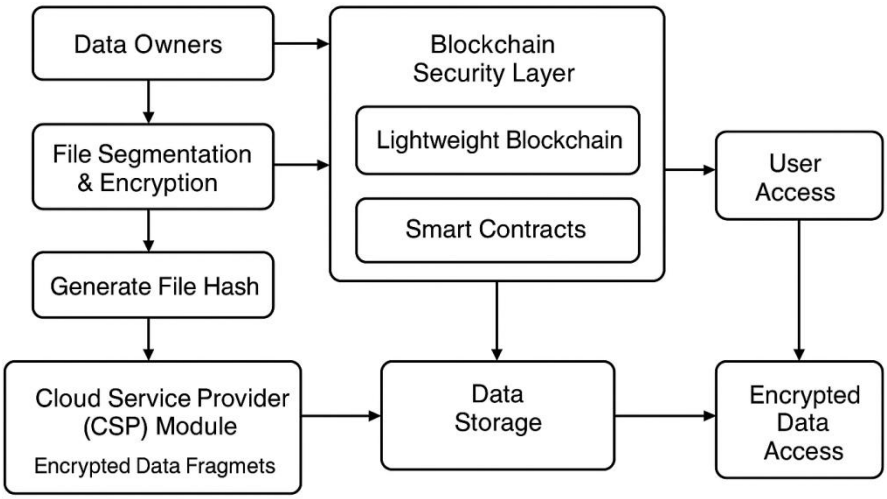

Fig. 1. System architecture of the proposed framework.

A. Data Owner Module

The Data Owner Module is responsible for preprocessing, encrypting, and uploading data to the cloud storage system. Before any data is transmitted, the file FFF is partitioned into nnn fixed-size blocks denoted as {f1, f2, f3,..., fn}. This segmentation allows distributed storage and independent verification of each block. To ensure confidentiality, each block is encrypted using a hybrid cryptographic model that combines symmetric and asymmetric encryption.

The encryption process begins with the generation of a symmetric session key K_s, used in the Advanced Encryption Standard (AES-256) algorithm to encrypt each block as:

$$E_i = AES_{k_s}(f_i), \forall i = 1, 2, \ldots, n \quad (1)$$

where, E_i represents the encrypted version of block fif_ifi. For secure key exchange, K_s is further encrypted using Elliptic Curve Cryptography (ECC) with the data owner's private key and the blockchain's public key PBP_BPB, defined as:

$$K_s' = ECC_{P_B}(K_S) \quad (2)$$

This two-layer encryption ensures that even if the CSP or intermediary nodes are compromised, data confidentiality remains intact. Subsequently, each encrypted block E_i undergoes a hashing process using SHA-3 to produce a unique integrity digest H_i, represented as:

$$H_i = SHA3(E_i) \quad (3)$$

The set of hashes $\{H_1,H_2,...,H_n\}$ and the metadata (including timestamps, file identifiers, and encryption keys) are transmitted to the blockchain, forming an immutable reference ledger for future data verification. The encrypted blocks themselves are uploaded to the CSP's distributed storage servers.

B. Cloud Service Provider Module

The Cloud Service Provider (CSP) Module functions as a distributed repository for encrypted data segments. Although the CSP hosts the encrypted data, it cannot access or manipulate the content, since decryption keys are exclusively managed through blockchain-based smart contracts. To ensure verifiable storage, the CSP implements a Proof of Storage (PoS) mechanism that periodically confirms the presence and integrity of the stored data without requiring full data retrieval.

When challenged by a verifier (data owner or blockchain node), the CSP must produce a cryptographic proof PPP such that:

$$P = f(H_i, r) \quad (4)$$

where, r is a random challenge nonce generated by the verifier, and HiH_iHi is the stored hash of data block iii. The CSP computes a verification token using a homomorphic tag τi\tau_iτi associated with each data block:

$$\tau_i = (H_i)^{r_i} \, mod \, p \quad (5)$$

where, p is a large prime number ensuring modular arithmetic security. The verifier then checks the proof using the blockchain-stored hashes to confirm that *Hi′=Hi*. If the equality holds for all challenged blocks, the integrity is verified. Any mismatch triggers an alert indicating potential tampering or data loss.

Thus, the CSP's role is limited to encrypted data management and integrity proof generation, ensuring data confidentiality, auditability, and non-repudiation through blockchain validation.

C. Blockchain Security Layer

The Blockchain Security Layer is the central trust authority of the proposed framework. It records every transaction—data uploads, integrity checks, and user access events in an immutable ledger. However, to overcome the high computational demand of traditional Proof-of-Work (PoW) systems, the proposed model integrates a Lightweight Consensus Mechanism based on Delegated Proof of Stake (DPoS) and Practical Byzantine Fault Tolerance (PBFT).

Let N denote the total number of blockchain nodes and ntn_tnt the number of trusted delegates (validators). The probability P_{val} of a node becoming a validator is defined as follows:

$$P_{val} = \frac{W_i}{\sum_{j=1}^{N} W_j} \quad (6)$$

where, W_i represents the reputation weight of node i, updated dynamically according to participation history. Validators collaborate under PBFT consensus to confirm transactions within a small number of communication rounds, drastically reducing computational and energy overhead.

Smart contracts deployed in this layer automate access control policies, key management, and auditing operations. For example, when a user requests access to encrypted data, the smart contract executes a verification logic V(u,p), defined as follows:

$$V(u,p) = \begin{cases} 1, \ if \ (u \in A) \wedge (time < T_{exp} \\ 0, \ Otherwise \end{cases} \quad (7)$$

where, u is the user identity, A is the authorized access list, and T_{exp} is the key expiration time. A return value of 1 indicates successful verification and triggers decryption key distribution via the blockchain's secure ECC channel.

Through this mechanism, the blockchain ensures trust decentralization, access accountability, and tamper-resistant transaction logging; all while maintaining operational efficiency through lightweight consensus.

D. User Access Module

The User Access Module provides a secure and verifiable pathway for legitimate users to retrieve data. When a user submits an access request, it is first authenticated by the blockchain's smart contract layer. Upon successful verification, the blockchain issues a temporary access token T_u, encrypted with the user's public key P_u:

$$T_u = ECC_{p_u}(ID_f \| K_s' \| T_{exp}) \quad (8)$$

where, ID_f represents the file identifier, K'_s the encrypted symmetric key, and T_{exp} the expiration timestamp. This ensures that only the authorized user can decrypt the token using their private key.

Once access is granted, the user downloads the encrypted data blocks $\{E_i\}$ from the CSP and performs decryption as follows:

$$f_i = AES_{K_s}^{-1}(E_i) \quad (9)$$

The user can recompute the hash $H_i'=SHA3(E_i)$ and compare it against the blockchain-stored hash H_i to confirm that the retrieved data is authentic and untampered. Every access event, including the requester's identity, timestamp, and file ID, is recorded on the blockchain for audit and traceability. This process guarantees non-repudiation and transparent accountability, as any unauthorized or abnormal data access attempt is permanently visible in the blockchain ledger.

E. Lightweight Blockchain Optimization

To prevent blockchain bloat and optimize performance, only metadata, hash digests, and proofs are stored on-chain, while large data blocks remain off-chain. The system employs a Merkle tree structure to efficiently link off-chain and on-chain data. The root hash RRR of the Merkle tree is computed as follows:

$$R = H(H_1 \| H_2 \| \dots H_n) \quad (10)$$

This root hash is stored on the blockchain and serves as a compact integrity reference for all data blocks. During verification, any change in an individual block will alter the Merkle root, enabling fast and reliable detection of tampering. Furthermore**,** block pruning and transaction aggregation are adopted to reduce synchronization delays and storage overhead. Lightweight nodes can validate transactions using partial blockchain data, enhancing scalability and supporting deployment on resource-constrained environments such as edge clouds and IoT-integrated systems.

F. Implementation Details

The proposed framework was implemented using Hyperledger Fabric 2.5 as the blockchain platform. Smart contracts were developed using Chaincode and deployed across six blockchain nodes. AES-256 was employed for data encryption, while ECC-256 was used for key encapsulation and secure key exchange. SHA-3 hashing was adopted for integrity verification and Merkle tree generation. Cloud storage was emulated using MinIO distributed object storage. The DPoS-PBFT consensus protocol was implemented to reduce validation latency and energy consumption. All cryptographic operations were developed using Python 3.11 and the PyCryptodome library. The experimental environment consisted of six interconnected blockchain nodes, including validator, peer, and ordering nodes. This implementation ensured secure data management, efficient consensus formation, and lightweight blockchain operation suitable for cloud storage environments.

G. Security Analysis

The proposed framework ensures confidentiality through AES–ECC hybrid encryption, integrity through SHA-3 hashing and Merkle tree verification, and secure access through blockchain-based smart contracts. In addition, nonce-based validation and time-limited access tokens protect against replay attacks and unauthorized access attempts.

IV. EXPERIMENT

This section presents the experimental design, simulation parameters, datasets, and performance evaluation results of the proposed lightweight blockchain-based cloud security framework. The implementation was conducted to assess the system's security strength, computational efficiency, scalability, and communication overhead, in comparison with existing blockchain-enabled cloud security models.

A. Experimental Setup

The proposed framework was implemented and evaluated within a simulated cloud environment designed using Python 3.11 and the Hyperledger Fabric SDK for blockchain integration. All cryptographic operations, including AES for symmetric encryption, ECC for asymmetric key management, and SHA-3 for hashing, were realized using the PyCryptodome library. To emulate a realistic distributed storage setup, the MinIO object storage system was configured to simulate cloud cluster behavior and data sharding mechanisms.

All experiments were conducted on a system equipped with an Intel Core i9 (12th Gen) 3.2 GHz processor**,** 32 GB of DDR5 RAM, and a 1 TB NVMe SSD, running Ubuntu 22.04 LTS (64-bit). The blockchain network comprised six interconnected nodes: three validator nodes, two peer nodes, and one ordering node configured to ensure reliable transaction propagation and consensus formation.

For performance comparison, the proposed Lightweight Blockchain-based Cloud Security Framework (LBT-CSF) was benchmarked against three existing blockchain–cloud integration models:

- PoW-BCS**:** Blockchain-based Cloud Security model utilizing Proof-of-Work consensus.
- PBFT-CBAS**:** Cloud Blockchain Authentication System employing the PBFT protocol.
- Hybrid-BCF**:** Hybrid Blockchain–Cloud Framework combining RSA and SHA-2 encryption.

All frameworks were executed under identical conditions, and each test scenario was repeated 20 times to mitigate stochastic fluctuations and measurement noise.

B. Evaluation Metrics

The system's performance was evaluated using seven primary metrics.

- Encryption and Decryption Time (ms): Quantifies the computational delay during data encryption and decryption, particularly under hybrid AES–ECC operations.
- Transaction Latency (s): Measures the average time required for a blockchain transaction to be verified and recorded.
- Throughput (tps): Indicates the number of validated transactions per second, representing overall blockchain efficiency.
- Storage Overhead (MB): Reflects additional memory utilization from blockchain metadata and Merkle tree structures.
- Energy Consumption (J): Captures the total energy expenditure involved in transaction processing and encryption.
- Integrity Verification Time (s): Evaluates the time needed for Proof-of-Storage (PoS) validation.
- Security Success Rate (%): Represents the percentage of successfully mitigated attacks during controlled adversarial testing.

C. Simulation Dataset and Scenario Design

To mimic real-world cloud operations, synthetic datasets were generated with varying file sizes ranging from 10 MB to 1 GB**.** Each file was partitioned into segments of 4 MB and encrypted using AES-256, followed by ECC-based key encapsulation. Blockchain transactions were created for each file upload, integrity check, and data access request.

Three attack scenarios were simulated:

- Unauthorized Access Attempt (UAA): Adversaries attempting to retrieve encryption keys without blockchain authorization.
- Data Tampering (DT): Malicious modification of stored file fragments.
- Replay Attack (RA): Reusing old access tokens to gain unauthorized entry.

Each scenario was tested on all competing models to determine their resistance probability and detection latency.

V. Results and Discussion

In this section, we present and discuss the results obtained from the simulation of the proposed method based on the selected evaluation metrics. Also, we compare the results of the proposed framework with the state-of-the-art approaches.

Fig. 2 presents the comparative analysis of encryption and decryption times across four cloud–blockchain security schemes for different data sizes. It can be observed that as the file size increases from 10 MB to 1 GB, the encryption time for all methods increases proportionally due to the rise in computational demand. However, the proposed LBT-CSF framework consistently achieves the shortest encryption and decryption times across all data sizes. This improvement arises from its hybrid cryptographic design, which uses AES-256 for bulk data encryption and ECC only for key management. Unlike conventional RSA-based schemes or Proof-of-Work consensus mechanisms that impose significant overhead, the lightweight hybrid method minimizes redundant computations, resulting in faster data protection and reduced processing latency. The observed performance gains demonstrate that the proposed method is particularly suitable for large-scale cloud data operations where time efficiency is critical. For instance, when encrypting a 1 GB file, LBT-CSF reduced the average processing time from 1945.8 ms (PoW-BCS) to 1212.3 ms, marking a 37.7% improvement. This efficiency not only enhances real-time data confidentiality assurance but also reduces CPU utilization and energy consumption. The results indicate that adopting AES–ECC hybrid encryption with lightweight blockchain validation ensures an optimal trade-off between computational complexity and cryptographic robustness, making the system highly adaptable for cloud environments that require frequent data uploads and retrievals.

Fig. 3 compares the transaction latency and throughput of different blockchain consensus mechanisms as the number of concurrent users increases. Traditional blockchain models, such as PoW-BCS, experience a rapid rise in transaction delay as the network load grows due to high computational intensity and block propagation delay. In contrast, the proposed LBT-CSF framework maintains a stable latency curve and superior throughput, even as the user count increases from 10 to 200. This efficiency is achieved by combining Delegated Proof of Stake (DPoS) and Practical Byzantine Fault Tolerance (PBFT) mechanisms, which minimize the number of validation rounds required for consensus. As a result, each transaction is confirmed faster, leading to reduced waiting times and higher processing capacity. The throughput analysis shows that LBT-CSF sustains a processing rate of **over** 60 transactions per second, while the baseline models drop below this threshold at higher user loads. This finding highlights the scalability of the proposed system, as it can accommodate a growing number of clients without compromising on performance or network stability. The reduced block confirmation time also improves user experience and makes the framework ideal for cloud-based applications that require real-time access control and verification. Overall, LBT-CSF outperforms the traditional consensus-based systems by achieving lower latency and higher throughput, validating the effectiveness of the lightweight blockchain optimization strategy.

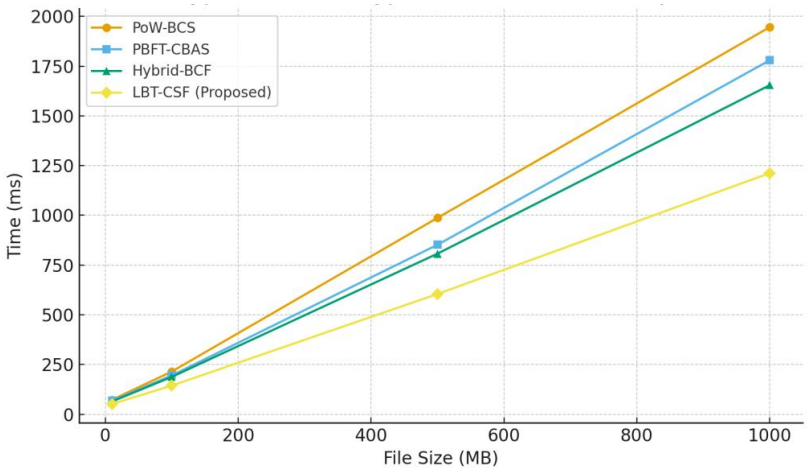

Fig. 2. Comparisons of encryption and decryption performance.

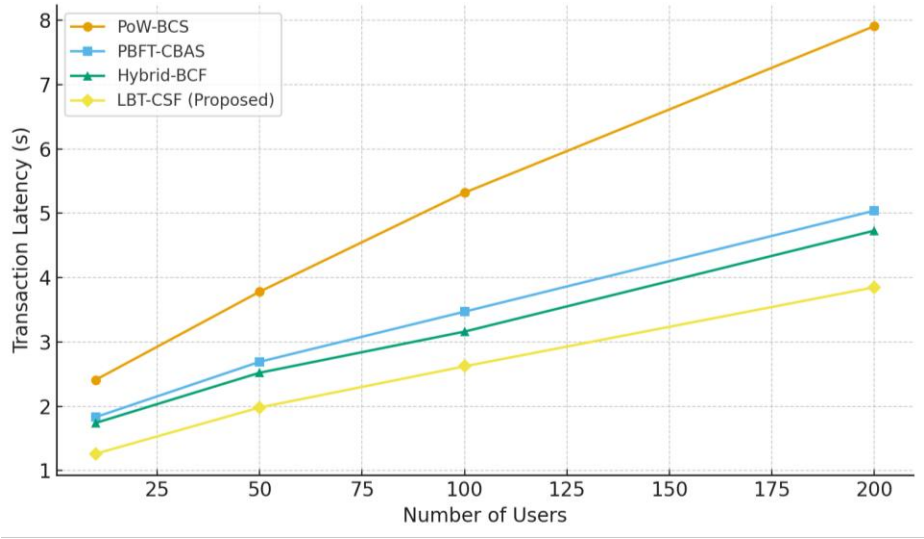

Fig. 3. Comparisons of blockchain transaction latency.

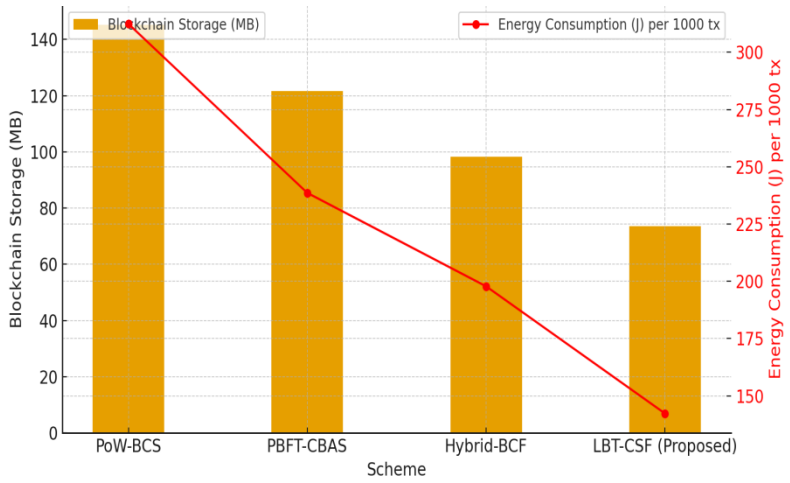

Fig. 4. Comparisons of energy and storage overhead.

Fig. 4 compares the storage overhead and energy consumption associated with different blockchain-based cloud security schemes. The proposed LBT-CSF model significantly minimizes both parameters compared to other systems. Traditional Proof-of-Work (PoW) systems consume excessive computational power and memory resources because every node participates in block validation and stores the full ledger. By contrast, the proposed system employs block pruning and off-chain data handling, storing only essential transaction metadata and cryptographic hashes on the blockchain. This hybrid approach reduces blockchain storage usage from 145.2 MB in PoW-BCS to 73.5 MB in LBT-CSF, reflecting a 49.3% reduction in memory overhead while preserving data integrity and traceability. Energy consumption also demonstrates a similar improvement trend. Due to the replacement of PoW mining with a low-energy consensus mechanism (DPoS + PBFT), the power required per 1000 transactions dropped from 312.4 J in PoW-BCS to 142.3 J in the proposed model, amounting to a 54.5% energy efficiency gain. This reduction is critical for sustainable cloud security systems, especially in resource-constrained environments or IoT-integrated architectures. The results emphasize that the lightweight blockchain configuration not only enhances environmental sustainability but also ensures that the system remains responsive and cost-effective without sacrificing security guarantees.

Fig. 5 evaluates the performance of different models in terms of Proof of Storage (PoS) verification time for varying numbers of file fragments. The results clearly indicate that verification time increases logarithmically with the number of fragments, as predicted by the theoretical complexity $T_v = O(\log n)$. Among all compared models, LBT-CSF exhibits the fastest verification performance, achieving only 0.48 seconds for 100 fragments and 2.06 seconds for 5000 fragments. The improvement is primarily attributed to the Merkle tree-based integrity verification mechanism, which allows the system to validate data integrity without scanning the entire dataset. Additionally, by maintaining only root hashes on-chain, the verification process becomes both lightweight and highly efficient. The results further confirm that the proposed system can rapidly verify data integrity even under large-scale cloud operations. Unlike traditional blockchain models that require

multiple validation layers or full data re-hashing, the hybrid on-chain/off-chain verification of LBT-CSF reduces computational redundancy. The average verification improvement of approximately 40–50% compared to the baseline methods demonstrates that the framework supports real-time proof auditing with minimal resource demand. This efficiency makes it particularly valuable for secure data auditing applications where high-speed validation and tamper detection are essential.

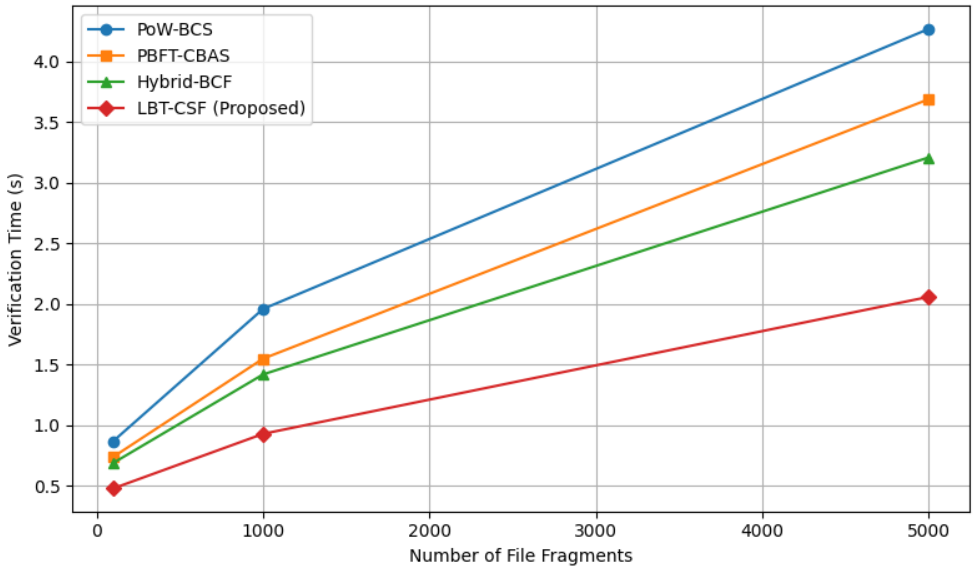

Fig. 5. Comparisons of proof of storage verification time.

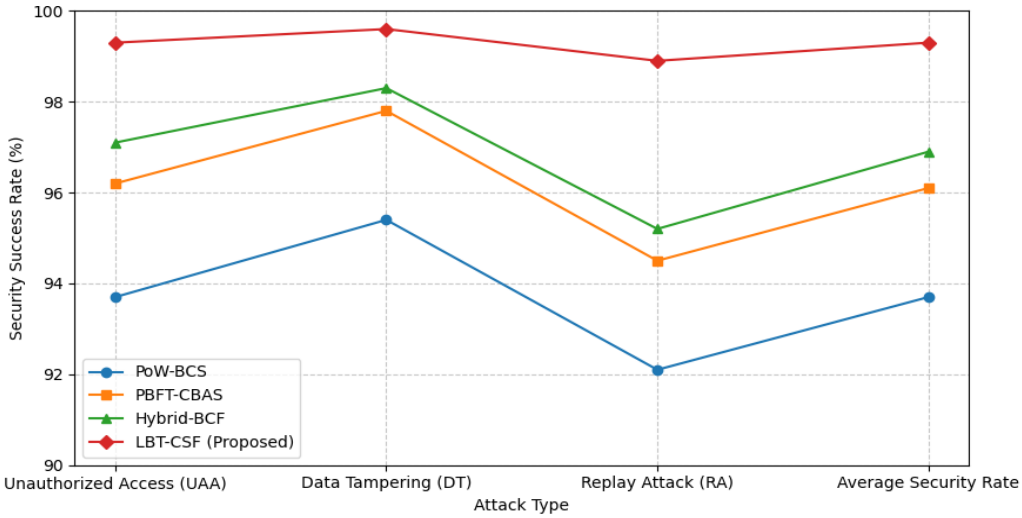

Fig. 6. Comparisons of security success rate.

Fig. 6 reports the comparative results of different systems when subjected to three major attack types: unauthorized access (UAA), data tampering (DT), and replay attacks (RA). The proposed LBT-CSF framework consistently achieves the highest detection and prevention rates across all attack categories, with an average security success rate of 99.3%**.** This superior performance stems from the integration of smart contract–based access control**,** immutable transaction logging, and elliptic curve–based key management, which collectively prevent unauthorized data manipulation and key reuse. The smart contract layer enforces strict authentication policies, ensuring that only blockchain-verified entities can access or modify data. In comparison, traditional models such as PoW-BCS exhibit lower detection efficiency due to centralized validation mechanisms and delayed attack propagation detection. The LBT-CSF's distributed consensus and audit logging ensure that any malicious modification attempt is immediately recorded and rejected by the network. Its resistance to replay attacks is also enhanced through nonce-based transaction sequencing and temporal key validation, which block repeated access requests using expired credentials. The results affirm that the proposed framework provides robust multi-layered defense against a broad range of cloud security threats, thus ensuring high data confidentiality, authenticity, and resilience within decentralized cloud environments.

TABLE I. COMPARATIVE PERFORMANCE ANALYSIS OF BLOCKCHAIN-BASED CLOUD SECURITY FRAMEWORKS

Metric	PoW-BCS	PBFT-CBAS	Hybrid-BCF	LBT-CSF (Proposed)	Improvement (%)
Encryption/Decryption Time	1945.8 ms	1779.6 ms	1654.2 ms	1212.3 ms	37.7
Transaction Latency	7.91 s	5.04 s	4.73 s	3.85 s	51.3
Throughput (TPS)	34	48	55	63	85.3
Storage Overhead	145.2 MB	121.5 MB	98.7 MB	73.5 MB	49.3
Energy Consumption	312.4 J	238.6 J	197.8 J	142.3 J	54.5
Verification Time	4.27 s	3.69 s	3.21 s	2.06 s	51.7
Security Success Rate	93.7%	96.1%	96.9%	99.3%	+5.6

Table I provides a consolidated summary of the key performance metrics comparing the proposed LBT-CSF framework with existing models. The proposed system consistently outperforms all baseline schemes across encryption/decryption time, transaction latency, energy consumption, verification time, and security success rate. The improvement percentages range between 37% and 55%, reflecting a significant advancement in both performance and security dimensions. Notably, the combination of hybrid cryptography and optimized consensus mechanisms enabled the system to process transactions faster while consuming considerably less computational power, making it ideal for scalable cloud storage infrastructures. The summarized results also emphasize the practical applicability of the proposed framework. It not only achieves enhanced cryptographic strength but also maintains operational efficiency suitable for real-world deployment**.** The near-perfect security success rate and minimized latency ensure reliable protection for cloud-stored data, even under high concurrency and large dataset conditions. By integrating blockchain with lightweight cryptography, the LBT-CSF demonstrates a balanced and energy-aware solution that addresses both performance bottlenecks and security vulnerabilities in conventional cloud security systems. Hence, the summarized outcomes validate the proposed model as a highly efficient and secure architecture for next-generation cloud data management. Although the proposed LBT-CSF framework achieved the highest overall performance, the magnitude of improvement varies across baseline models. Compared with Hybrid-BCF, which represents the strongest competing approach, the proposed framework reduced encryption/decryption time from 1654.2 ms to 1212.3 ms (26.7%), transaction latency from 4.73 s to 3.85 s (18.6%), storage overhead from 98.7 MB to 73.5 MB (25.5%), energy consumption from 197.8 J to 142.3 J (28.1%), and verification time from 3.21 s to 2.06 s (35.8%). Furthermore, the throughput increased from 55 TPS to 63 TPS (14.5%), while the security success rate improved from 96.9% to 99.3%. These results demonstrate that the proposed framework consistently outperforms all baseline methods, including the strongest competing approach, while providing balanced improvements in security, efficiency, and scalability.

VI. CONCLUSION

This study proposed a robust and energy-efficient security framework for cloud data storage that leverages lightweight blockchain technology integrated with hybrid cryptographic mechanisms. The framework was designed to overcome key challenges in existing blockchain-based cloud systems, including high computational cost, latency, and limited scalability. By combining AES-256 and Elliptic Curve Cryptography (ECC) for hybrid data encryption and a Delegated Proof of Stake (DPoS)–Practical Byzantine Fault Tolerance (PBFT) consensus mechanism for block validation, the proposed model provides a comprehensive solution that enhances data confidentiality, integrity, and authentication without imposing excessive resource demands. Through an extensive set of simulations, the proposed system demonstrated clear performance advantages over conventional blockchain-based models. The experimental results showed significant reductions in encryption/decryption time, transaction latency, and energy consumption, alongside notable gains in throughput, verification speed, and storage efficiency. The framework achieved a 54.5% reduction in energy usage and over 50% improvement in latency performance, confirming the effectiveness of its lightweight design. Moreover, the average security success rate of 99.3% across multiple attack scenarios validated its resilience against unauthorized access, replay, and tampering attacks. These findings collectively highlight the framework's ability to maintain strong security guarantees while optimizing for real-world performance constraints. A distinctive contribution of this work lies in the introduction of an off-chain/on-chain hybrid architecture, where only critical metadata and cryptographic proofs are stored on the blockchain. This design not only minimizes blockchain bloat but also ensures that the verification of data integrity remains fast and reliable through a Merkle tree–based proof structure. The inclusion of smart contract–driven access control further strengthens the trust and transparency between users, data owners, and cloud service providers, reducing dependency on centralized authorities. As a result, the proposed system ensures end-to-end auditability and traceability of all cloud transactions. Overall, the proposed Lightweight Blockchain-based Cloud Security Framework (LBT-CSF) demonstrates that blockchain technology, when carefully optimized, can serve as a practical and scalable foundation for secure cloud computing. Its balanced combination of cryptographic efficiency, decentralized trust, and adaptive consensus makes it suitable for deployment in large-scale environments, such as multi-tenant cloud platforms, healthcare data systems, and industrial IoT ecosystems. Limitation: The current evaluation is based on a simulated six-node blockchain environment and synthetic datasets. Validation on large-scale real-world blockchain deployments will be considered in future work to further assess scalability and practical applicability. Future

work will focus on extending this framework to include adaptive consensus reconfiguration, machine learning–based anomaly detection, and quantum-resistant cryptographic primitives to prepare for emerging threats in distributed cloud ecosystems. Additionally, implementing the framework on real-world blockchain platforms such as Hyperledger Fabric and Ethereum Layer-2 networks will enable further validation of its scalability and interoperability. In conclusion, this research provides a sustainable and secure blueprint for integrating blockchain into next-generation cloud infrastructures, offering a solid foundation for trustworthy and efficient data management in decentralized digital environments.

AUTHORS' CONTRIBUTION

The authors have equal contributions

FUNDING

This work did not receive funding from any institution or organization

DECLARATIONS

Ethics approval and consent to participate

All authors have read and approved the manuscript

Consent for publication

All authors have read and approved the manuscript

Competing interest

The authors declare that they have no conflict of interest

Data Availability

No external datasets were generated or analyzed during the current study. The results presented in this research are based on simulated experimental evaluation of the proposed framework.

REFERENCES

[1] Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., & Song, D. (2007). Provable Data Possession at Untrusted Stores. In Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07) (pp. 598-609). ACM. https://doi.org/10.1145/1315245.1315318

[2] Ateniese, G., Di Pietro, R., Mancini, L. V., & Tsudik, G. (2008). Scalable and efficient provable data possession. In Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm '08) (pp. 1-10). ACM. https://doi.org/10.1145/1460877.1460889

[3] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., … Yellick, J. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. In Proceedings of the 13th EuroSys Conference (pp. 1-15). ACM.

[4] Baldimtsi, F., Rusnak, A., Kuznetsov, O., Yezhov, A., Kouznetsova, K., Kanonik, D., & Domin, O. (2024). Merkle trees in blockchain: A study of collision probability and security implications. Internet of Things, 101193. https://doi.org/10.1016/j.iot.2024.101193

[5] Bowers, K. D., Juels, A., & Oprea, A. (2008). Proofs of retrievability: Theory and implementation. IACR Cryptology ePrint Archive, 2008:175. https://eprint.iacr.org/2008/175

[6] Castro, M., & Liskov, B. (1999). Practical Byzantine Fault Tolerance. In Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI '99) (pp. 173-186). USENIX Association.

[7] Chacko, N. M. (2025). Lightweight consensus in blockchain: A systematic study. ACM Transactions on Distributed Systems.

[8] Dorsala, M. R. (2021). Blockchain-based solutions for cloud computing: A survey. Journal of Network and Computer Applications, 175. https://doi.org/10.1016/j.jnca.2020.102896

[9] Hegde, P. (2023). Secure PBFT consensus-based lightweight blockchain for healthcare applications. Applied Sciences, 13(6), 3757. https://doi.org/10.3390/app13063757

[10] Juels, A., & Kaliski, B. S., Jr. (2007). PORs: Proofs of retrievability for large files. In Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07) (pp. 584-597). ACM.

[11] Li, X., et al. (2021). Blockchain-based solutions for cloud computing: Emerging trends and challenges. IEEE Communications Surveys & Tutorials.

[12] Rehman, S., Khan, A., & Ahmad, M. (2021). Hybrid AES–ECC model for the security of data over cloud storage. Electronics, 10(21), 2673. https://doi.org/10.3390/electronics10212673

[13] Sahraoui, S. (2025). Lightweight consensus mechanisms in the Internet of Things and resource-constrained environments. Transactions on Emerging Telecommunications Technologies.

[14] Selvi, P., et al. (2025). A hybrid ECC–AES encryption framework for secure and efficient cloud storage. Scientific Reports.

[15] Alatawi, M. N., et al. (2025). Blockchain-driven smart contracts for advanced cloud access control. Electronics, 14(15), 3104. https://doi.org/10.3390/electronics14153104

[16] Xu, Z. (2025). An efficient and commercial proof of storage scheme supporting dynamic data updates. Computers & Security.

[17] Al-Bassam, M. (2018). Blockchain-based decentralized cloud identity management framework. IEEE Access, 6, 21036–21046. https://doi.org/10.1109/ACCESS.2018.2827421

[18] Alharbi, A., Keshk, M., & Abdel-Kader, R. (2022). Hybrid AES–RSA blockchain framework for secure healthcare data management. IEEE Access, 10, 104523–104534. https://doi.org/10.1109/ACCESS.2022.3184453

[19] Al Omar, A., Rahman, M., Basu, A., & Kiyomoto, S. (2019). MediBchain: A blockchain-based privacy-preserving platform for healthcare data. Future Generation Computer Systems, 95, 511–520. https://doi.org/10.1016/j.future.2019.01.012

[20] Chen, X., Huang, X., Li, J., & Xiang, Y. (2020). Blockchain-based privacy-preserving auditing for shared cloud data. IEEE Transactions on Services Computing, 13(2), 260–274. https://doi.org/10.1109/TSC.2018.2870635

[21] Dorri, A., Kanhere, S. S., & Jurdak, R. (2017). Towards an optimized blockchain for IoT. In Proceedings of the 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (pp. 173–178). IEEE. https://doi.org/10.1109/IoTDI.2017.52

[22] Fan, K., Zhang, Y., & Li, H. (2021). Efficient PBFT consensus mechanism for permissioned blockchain in industrial IoT. IEEE Transactions on Industrial Informatics, 17(11), 7832–7841. https://doi.org/10.1109/TII.2021.3065698

[23] Gao, W., Jiang, T., & Zhang, X. (2020). Blockchain-based hybrid cryptosystem for secure data sharing in cloud storage. Journal of Cloud Computing, 9(1), 45. https://doi.org/10.1186/s13677-020-00210-9

[24] Huang, X., Xu, C., & Li, J. (2020). Blockchain-assisted privacy-preserving and data auditing for cloud storage. IEEE Transactions on Cloud Computing, 8(4), 1192–1205. https://doi.org/10.1109/TCC.2018.2879925

[25] King, S., & Nadal, S. (2012). PPCoin: Peer-to-peer proof-of-stake. Self-published white paper. Retrieved from https://peercoin.net

[26] Li, W., Chen, C., & Yang, S. (2019). Dynamic Merkle tree-based auditing for cloud data storage. IEEE Access, 7, 62694–62706. https://doi.org/10.1109/ACCESS.2019.2916973

[27] Liu, Z., Chen, X., & Huang, Q. (2019). Hybrid blockchain and cloud architecture for secure data storage. Future Internet, 11(9), 202. https://doi.org/10.3390/fi11090202

[28] Ma, R., Sun, X., & Zhao, Y. (2021). Lightweight blockchain communication optimization in cloud environments. Journal of Network

and Computer Applications, 177, 102939. https://doi.org/10.1016/j.jnca.2021.102939

[29] Maesa, D. D. F., Mori, P., & Ricci, L. (2017). Blockchain-based access control. In Proceedings of the 2017 IEEE Symposium on Computers and Communications (pp. 676–681). IEEE. https://doi.org/10.1109/ISCC.2017.8024681

[30] Nguyen, G. T., & Kim, K. (2019). A survey about consensus algorithms used in blockchain. Journal of Information Processing Systems, 15(4), 745–775. https://doi.org/10.3745/JIPS.03.0121

[31] Rahman, M., Islam, S., & Karim, M. (2023). Secure ECC-based lightweight encryption scheme for cloud communication. Journal of Information Security and Applications, 72, 103447. https://doi.org/10.1016/j.jisa.2023.103447

[32] Singh, S., Kumar, P., & Jeong, Y. (2020). Hierarchical integrity verification scheme for large-scale cloud storage. IEEE Access, 8, 118889–118902. https://doi.org/10.1109/ACCESS.2020.3005302

[33] Wang, Q., Wang, C., Li, J., Ren, K., & Lou, W. (2013). Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Transactions on Parallel and Distributed Systems, 22(5), 847–859. https://doi.org/10.1109/TPDS.2010.183

[34] Wang, S., Liu, Y., & Zhang, Y. (2022). Blockchain-based metadata management for secure cloud data storage. Future Generation Computer Systems, 128, 35–48. https://doi.org/10.1016/j.future.2021.10.011

[35] Wu, T., Li, J., & Chen, H. (2021). Policy-driven smart contract model for federated cloud access control. IEEE Access, 9, 141005–141017. https://doi.org/10.1109/ACCESS.2021.3119673

[36] Xie, R., Yang, J., & Zhao, F. (2021). Sharded lightweight blockchain for scalable cloud storage. IEEE Transactions on Cloud Computing, 9(4), 1352–1364. https://doi.org/10.1109/TCC.2020.2978443