

# Blockchain Integrated Distributed Threshold Elliptic Curve Cryptography Framework for Scalable and Secure Cloud Data Storage

Naga Swetha Dhulipudi and Balaji Savadam

*Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad, India*

## Article history

Received: 07-10-2025

Revised: 11-08-2026

Accepted: 21-09-2026

## Corresponding Author:

Naga Swetha Dhulipudi  
Department of Computer  
Science and Engineering,  
Koneru Lakshmaiah Education  
Foundation, Hyderabad, India  
Email:  
mailsforswetha@klh.edu.in

**Abstract:** The development of cloud computing offers scalable processing capabilities, thereby enabling organizations to store and process sensitive data in the cloud. However, it faces crucial security challenges, as existing cryptographic and blockchain systems suffer from poor scalability and centralized key management, which creates a single point of failure and introduces vulnerabilities to both quantum and classical attacks. This research proposes a Distributed Threshold Elliptic Curve Cryptography (DT-ECC) framework integrated with blockchain to enhance security, resilience, and transparency. DT-ECC utilizes a threshold-based decentralized key generation mechanism, thereby eliminating a single point of failure and enhancing the security of cryptographic keys. Furthermore, Quantum Key Distribution (QKD) is integrated to ensure quantum-resilient key exchanges, while Merkle tree-based verification ensures tamper-proof data integrity. The DT-ECC provides robust key management, encryption, and decryption, thereby minimizing memory utilization and latency. Experimental results demonstrate that DT-ECC achieves a key generation time of 5.65 s and an encryption time of 5.90 s, respectively, for a data size of 30MB, thereby outperforming existing models in secure cloud data storage and transmission.

**Keywords:** Blockchain, Cloud Computing, Distributed Threshold Elliptic Curve Cryptography, Key Management, Merkle Tree, Quantum Key Distribution

## Introduction

The term “cloud computing” is used in computer science to refer to outsourcing computing services, including computing power, storage, and applications. These services are accessed by users without concern for the complexity of power production or delivery (Altahat et al., 2024). It enables users to take advantage of data storage, computing resources, and application development platforms without requiring strong expertise in the underlying infrastructure (Ojha et al., 2024). Cloud computing is becoming so ubiquitous that the term “cloud” is often used in network construction to refer to the Internet, though this oversimplifies its complexity (Mondal and Chatterjee, 2024). With the application of cloud-based computation and storage, individuals and businesses can reduce initial investment costs and streamline infrastructure management (Li et al., 2024). Internet of Things (IoT) devices are resource-limited and require real-time services, and large Cloud Service Providers (CSPs) provide numerous solutions to

guarantee Quality of Service (QoS) (Krishnan, 2024). The combination of fog and edge computing technologies enables CSPs to deliver more convenient and resource-saving services to cloud-based IoT users (Zubair and Ahmed, 2024; Gupta et al., 2024). Cloud computing services are usually categorized into three types: Public, private, and hybrid. Among these, the most popular is the public cloud because it is highly flexible and cost-efficient, with enhanced customer support, which is leading to its rapid adoption. Nonetheless, data safety in the cloud remains a vital issue, making data security and encryption a constant focus of research (Umar et al., 2024; Li et al., 2025).

Cloud data storage is considered one of the major services offered in the cloud computing field under Infrastructure as a Service (IaaS). As its popularity continues to rise due to its wide range of benefits, both users and service providers face increasing concerns regarding privacy and security (Dhamdhare et al., 2025). To counter these risks, encryption techniques, primarily including symmetric and asymmetric cryptography, are

widely adopted (Shakor et al., 2024). The difference between them lies in the number of keys involved: Symmetric cryptography uses a single key shared between two communicating parties, while asymmetric cryptography employs a pair of keys one for encryption and one for decryption. The use of longer and more intricate keys further strengthens encryption algorithms, making them more resistant to attacks (Yihui, 2024). Notwithstanding these efforts, several issues persist, including security breaches, transmission errors, overloads, channel noise, limitations of cloud infrastructure design, and unauthorized data transfers by malicious parties (Bhattacharjee et al., 2024; Chowdhury et al., 2025). Blockchain technology is also incorporated into smart healthcare systems to secure information, regulate information sharing, and control access to patient records, among other applications (Sultana and Srinivas, 2024). Another application of blockchain is in the architecture of cybersecurity for intelligent household networks in smart healthcare systems (Zhang et al., 2024; Kathole et al., 2024). Nevertheless, many established systems remain insufficient, as they pay limited attention to key security metrics and inadequately enforce protective mechanisms (Pradhan and Priyadarsini, 2024). In addition, conventional blockchain models are hindered by high energy consumption and poor scalability, limiting their integration into contemporary cloud environments (Liu et al., 2024). In the proposed model, Distributed Threshold Elliptic Curve Cryptography (DT-ECC) is responsible for distributed threshold-based key generation and secure data encryption. Quantum Key Distribution (QKD) is employed to provide quantum-resilient session keys among communicating entities, and the Merkle tree is utilized to verify the integrity of encrypted data chunks before and after cloud storage. Blockchain acts as a trust layer by recording key commitments, integrity proofs, and transaction logs in an immutable way. By using QKD for secure decentralized key shares, the system provides a high protection against both classical and quantum attacks while DT-ECC maintains operational efficiency.

### *Problem Statement*

Cloud computing is emerging as a central platform for storing and processing sensitive data across various fields. However, cryptographic and blockchain security face limitations such as high computational overhead, lack of scalability, centralized key management risks, and vulnerability to blockchain attacks. The single point of vulnerability arises with keys generated at a single central point, and conventional encryption algorithms remain exposed to classical and quantum-based attacks.

### *Objective*

The objective of this research is to develop a DT-ECC framework integrated with blockchain to provide secure, distributed key generation and administration. The DT-

ECC mitigates single points of failure and provides better auditability through immutable blockchain records. Moreover, it improves resistance to quantum-based attacks by integrating QKD. Additionally, the DT-ECC minimizes encryption time, decryption time, latency, and memory consumption, thereby providing an effective, scalable, and robust security mechanism for cloud data storage and transmission.

### *Research Gap and Novelty*

Existing cloud security approaches address cryptographic security, blockchain-based trust, and quantum-secure communication, but do not ensure decentralized key management, quantum-resilient key exchange, immutable auditing, and efficient integrity verification. To address this gap, the proposed DT-ECC integrates distributed threshold ECC, blockchain, QKD, and Merkle tree verification into one framework. This mitigates single points of failure, improves resilience against classical and quantum attacks, enhances scalability, and provides better computational efficiency than existing approaches.

### *Contributions*

The main contributions of this research are as follows:

- This research proposes DT-ECC integrated with blockchain, enabling decentralized key generation to eliminate single points of failure and enhance cloud data security
- The proposed DT-ECC integrates QKD for quantum-resilient key exchange and Merkle-tree-based verification for tamper-proof data integrity, thereby providing robustness and confidentiality against internal and external threats
- Simulation results demonstrate that DT-ECC outperforms existing methods in terms of key generation time, response time, encryption time, decryption time, memory utilization, and latency, making it lightweight and highly secure for cloud data storage and transmission

### *Literature Review*

Shrivastava et al. (2024) proposed a hybrid lightweight encryption framework based on a permissioned blockchain to secure cloud data sharing. The approach addressed traditional model issues such as unauthorized access and performance inefficiencies. By integrating elliptic-curve key generation, Flamingo Search Optimization (FSO), hybrid ElGamal encryption, and Secure Hash Algorithm-256 (SHA-256) hashing with proof-of-authority validation, this method unified encryption and integrity verification. Consequently, it reduced computational and storage overheads while

ensuring confidentiality and verifiable auditability. However, this approach remained inefficient in terms of adversarial robustness, key revocation, and scalability.

Ahmad and Mehruz (2024) presented a Time-Oriented Latency Approximation-based Data Encryption (TLADE) technique that dynamically selected both encryption algorithms and data transmission routes. Compared with conventional access-control approaches, TLADE combined cryptographic selection with network scheduling to preserve both security and responsiveness, which was critical for maintaining real-time service quality in cloud environments. While results showed reduced latency, improved throughput, and secure data exchange, TLADE did not quantify the trade offs between latency, cryptographic strength, and key management complexity, which limited its generalizability.

Jin (2024) introduced a blockchain enabled economic risk assessment framework to reinforce cloud data protection while supporting privacy-preserving analysis. The architecture combined fine-grained access control, multi-edge servers, and decentralized anonymity mechanisms with blockchain auditability to provide a verifiable and trustworthy workflow. The method supported secure, accountable data sharing across organizational boundaries, demonstrating strengthened privacy, reduced computational burden, and secure access control. Nonetheless, the system introduced architectural complexity and faced challenges related to collusion resilience, revocation, and scalability.

Bai et al. (2024) introduced Collusion-Resistant File Hierarchies Ciphertext-Policy Attribute-Based Encryption (CR-FH-CPABE), designed to reduce redundancy in multi-file cloud storage. By encoding hierarchical policy relationships, this approach compressed ciphertexts across related files, thereby reducing encryption and storage costs. It ensured fine-grained access control while resisting collusion among unauthorized users, making it especially relevant for cloud repositories with overlapping access structures. The model demonstrated improved confidentiality and storage efficiency. However, it did not address challenges related to key revocation, dynamic policy updates, and practical deployment.

Akbar et al. (2024) proposed a blockchain-based security trust model that integrated QKD with a Modified Advanced Encryption Standard (MAES) cipher for robust cloud data protection. This approach combined quantum-secured key exchange with tamper-evident blockchain records and efficient symmetric encryption to provide multi-layered defense. It was justified by the inadequacy of classical encryption against sophisticated adversaries. The results demonstrated stronger confidentiality, tamper resistance, and enhanced operational security. Nevertheless, challenges such as QKD deployment costs,

infrastructure constraints, and scalability in heterogeneous cloud networks remained significant.

Tang et al. (2024) addressed the challenge of privacy-preserving deduplication by adopting a hybrid cloud structure, where private clouds managed secure key generation and access control, while public clouds handled storage. By separating security management from data storage, the model ensured that deduplication did not compromise private information.

Specifically, this approach achieved integrity protection, secure access, and privacy during duplicate suppression, particularly beneficial for enterprise-level data management. It demonstrated enhanced confidentiality and practical usability. However, limitations such as storage overhead in private clouds, challenges in block-level deduplication, and vulnerability to emerging attack models remained.

Compared with existing QKD-blockchain approaches, the DT-ECC utilizes QKD for secure session key generation, while the cryptographic protection is provided by distributed threshold ECC. This provides QKD-based secure decentralized key shares, thereby differentiating the DT-ECC from existing methods. The DT-ECC integrates distributed threshold ECC for decentralized key management, blockchain for immutable trust, QKD for secure session key generation, and Merkle tree verification for encrypted cloud data integrity.

## Materials and Methods

This section provides the foundation of the research method designed to prioritize privacy and data security in cloud computing. The DT-ECC introduces a multi-layered defense architecture capable of mitigating diverse risks for cloud data storage and transmission. In this model, encryption keys are generated and managed among multiple nodes using a threshold mechanism, thereby eliminating single points of failure. Blockchain ensures tamper-proof auditability and immutability for all key management and data transactions.

Fig. 1 presents a block diagram of the proposed method. The cybersecurity trust model utilizes blockchain to maintain a transparent and tamper-resistant ledger, while the multi-risk protection model employs DT-ECC for decentralized key control and robust cryptographic protection of sensitive data. Furthermore, QKD is used to enable secure, physics-based key exchanges resilient to both classical and quantum attacks. Merkle tree-based verification is integrated to ensure data integrity. Overall, this framework provides an adaptive and comprehensive security mechanism that ensures secure data transmission and storage in cloud computing, thereby offering improved protection against cyber threats.

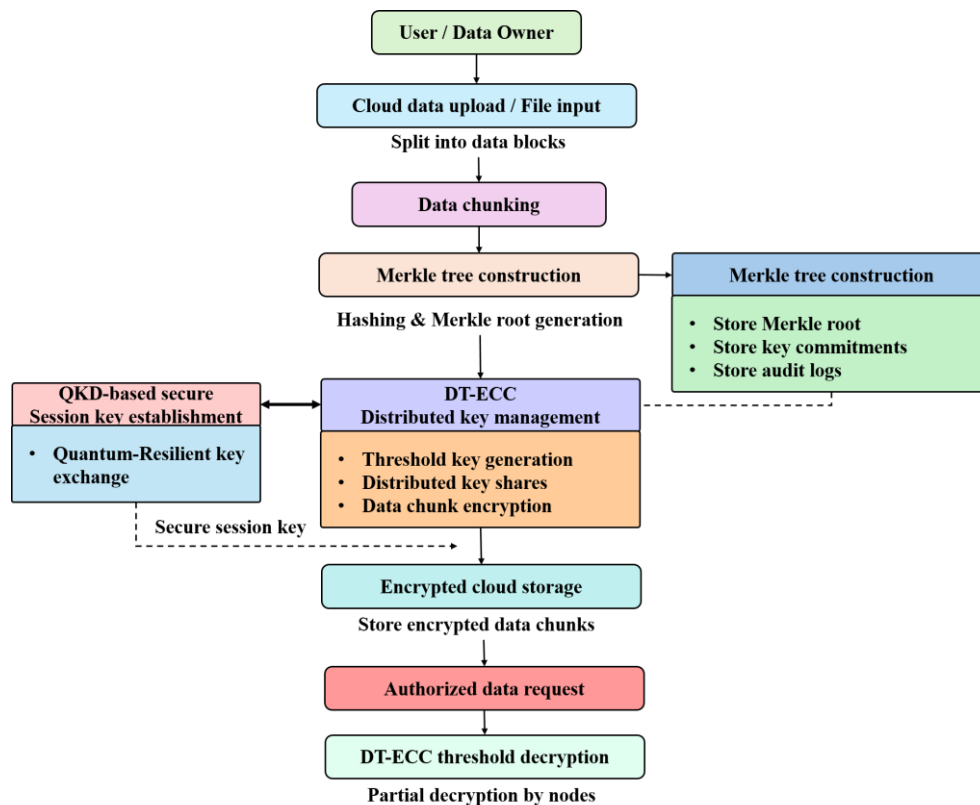


**Fig. 1:** Overall workflow of the proposed DT-ECC framework integrating blockchain, QKD, and Merkle tree verification for secure cloud data storage

The proposed framework operates in four stages: Secure key establishment, distributed threshold key generation, cloud data encryption/decryption, and integrity verification. The proposed framework contains four modules: Data processing, DT-ECC-based distributed key management, permissioned blockchain, and QKD-enabled secure communication. DT-ECC provides and reconstructs keys using threshold secret sharing, removing single points of failure. For DT-ECC, an ECC-based  $(t, n)$  threshold mechanism is utilized, where key shares are distributed among  $n$  authorized nodes and  $t$  valid shares are required for key reconstruction and decryption.

QKD provides secure key exchange, DT-ECC performs threshold-based cryptographic operations, blockchain maintains immutable transaction and key records, and Merkle trees verify data integrity during cloud data storage. The integration process operates in multiple coordinated stages. Initially, the input file is split into multiple data segments, and a Merkle tree is created by hashing each data segment and computing parent hashes until the Merkle root is attained. The root hash is recorded on the blockchain to provide immutable integrity. Then, QKD is used between communicating entities to create a quantum-resilient session key for

secure transmission of cryptographic parameters. Subsequently, DT-ECC performs distributed threshold key generation, where multiple nodes generate private key shares without revealing the whole secret to a single entity. The generated public key is utilized for secure encryption of cloud data. In the retrieval process, the required threshold of nodes performs partial decryption and recreates the plaintext securely. Lastly, the integrity of the recovered data is verified by estimating the Merkle root and comparing it with the blockchain-stored root hash. This integrated process ensures confidentiality through DT-ECC, secure key establishment by QKD, and tamper-proof integrity verification through Merkle trees. QKD is integrated as a conceptual security component in this research to provide quantum-resilient session keys between communicating nodes. It assumes secure QKD-based key establishment to estimate the incorporation with DT-ECC and blockchain, thereby providing assessment of a quantum-resilient architecture while maintaining a simulation-based implementation.

### Cybersecurity Trust Model

The cybersecurity trust model enhances privacy and data security in cloud computing by employing blockchain technology to create a secure and tamper-

resistant ledger for every cloud data transmission. The ledger ensures auditability, integrity, and traceability of stored and transmitted data. Its decentralized architecture distributes authentication and authorization across multiple nodes, eliminating single points of failure and reducing the risk of unauthorized access. Leveraging blockchain's transparency and immutability, the model provides a trust-based security framework that ensures confidentiality and integrity within cloud environments.

### Blockchain Technology

The blockchain-based cybersecurity trust model secures data storage and transmission in cloud computing through distributed ledger technology. Blockchain employs cryptographic techniques to link blocks of records continuously, where each block includes a cryptographic hash of the previous block, a timestamp, and transaction data.

The framework includes interconnected nodes representing individuals and organizations participating in the cloud network. Transactions form the basic units of the blockchain, representing secure data exchanges among participants. The Peer-to-Peer (P2P) network allows nodes to communicate and share data without centralized control.

Blocks function as structures containing multiple validated transactions, and the blockchain ledger maintains tamper-resistant records accessible to authorized participants. By integrating blockchain into cloud security, this model mitigates single points of failure, prevents unauthorized data manipulation, and promotes accountability, trust, and resilience in cloud data security.

### Multi Risk Protection Model

The multi-risk protection model improves cloud data security using a DT-ECC cryptographic framework integrated with QKD for an additional layer of protection. DT-ECC employs a threshold-based mechanism for decentralized key generation and management, ensuring that no single node manages all private keys. This mitigates single points of failure and enhances resistance to internal and external attacks. DT-ECC forms the primary encryption and decryption mechanism for securing sensitive cloud data.

To further reinforce security during key exchanges, QKD generates encryption keys based on quantum principles, making the process resilient to interception and quantum-based attacks. The proposed DT-ECC method also facilitates the integration of advanced key distribution techniques in the cloud, thereby ensuring a high level of security.

### Distributed Threshold Elliptic Curve Cryptography

The proposed DT-ECC is employed to encrypt and

decrypt sensitive data stored and transmitted in cloud computing environments, where cloud platforms typically rely on databases. Since the direct encryption output is not always suitable for storage, encrypted data (ciphertext) undergoes appropriate encoding to ensure accurate storage and retrieval.

Encryption processes can be broadly categorized as symmetric and asymmetric approaches. In asymmetric encryption, two keys are used: A private/special key and a public/common key. While secure, it is computationally intensive and less efficient for large files. In symmetric encryption, the process is executed in blocks, enabling faster data encryption. To overcome these limitations while retaining secure key exchange, ECC is applied to encrypt and decrypt complex cloud data, such as encryption keys, which exist as plaintext and must be protected prior to processing.

A private key is typically represented as a 256-bit integer generated randomly within a specific range on an elliptic curve, ensuring strong security. The remaining integers within this range are considered valid private keys. Public keys are represented as points with coordinates  $\{x, y\}$  on the curve. ECC utilizes elliptic curves, which are defined mathematically, as shown in Eq. 1:

$$y^2 = x^3 + v_1x + v_2 \quad (1)$$

In cloud data security, ECC operates over elliptic curves defined in a finite field  $F_p$ , where  $p$  is a prime number (e.g.,  $p > 3$ ), or in  $F_{2^m}$ , where the field size is  $p = 2^m$ . The finite field is presented as a discrete set of points, where each point on the elliptic curve is characterized by numerical coordinates defined within the selected field. All algebraic operations are performed on points within the same field, ensuring consistent cryptographic computation. The general elliptic curve equation over a finite field is given in Eq. 2:

$$y^2 = x^3 + v_1x + v_2(\text{mod } p) \quad (2)$$

In the proposed model, conventional ECC key generation is enhanced by applying DT key generation, which eliminates centralized trust. Here, the authorized cloud node  $n$  participates in the generation of an ECC private key without relying on a trusted dealer. The authorized cloud nodes are pre-registered as permissioned blockchain participants and authenticated through identity verification and cryptographic records. Only the verified nodes are considered to perform security operations, including distributed threshold key generation and partial decryption. Every node  $P_i$  provides a random polynomial  $f_i(x)$  of degree  $t - 1$  over  $F_p$ , from which the secret shares  $s_{ij} = f_i(j)$  are calculated and securely transmitted to peers (Shamir, 1979). Concurrently, every node uses

cryptographic commitments to blockchain smart contracts that act as public data (Pedersen, 2007). Once verified, the ECC public key is computed as  $X = \sum_{i=1}^n f_i(0) \cdot G$ , where  $G$  is an elliptic curve generator point, and the aggregated public key is stored in the cloud. The decryption process requires at least  $t$  partial computations from various nodes, thereby ensuring that key misuse is impossible without the threshold.

DT key generation in ECC enhances cloud security by eliminating reliance on a single trusted entity for key generation and storage. Instead of storing the private key in a single node, cryptographic shares are distributed among multiple nodes using verifiable secret sharing. Decentralization alleviates single points of failure, making it more difficult for attackers to compromise the key because at least  $t$  out of  $n$  nodes are required for reconstruction. Blockchain-based coordination of the key generation process ensures tamper-proof recording of key operations, including updates and revocations, thereby providing immutable auditability and nonrepudiation.

Moreover, DT key generation provides secure, distributed decryption operations without revealing the complete key, thereby mitigating the risk of leakage. In cloud environments, dynamic rekeying is supported when nodes are added or removed, thereby preserving forward secrecy and adaptability to threats. By using DT key generation in ECC, the model provides scalable, resilient, and transparent security for sensitive cloud data.

In Rivest-Shamir-Adleman (RSA) encryption and decryption, numbers in the range  $[0, p - 1]$  are used for generating keys, whereas ECC uses the points  $\{x, y\}$

within  $F_p$  to generate cryptographic keys. The point-based representation enables effective key generation and higher security with smaller key sizes compared to RSA, making ECC suitable for security-critical data stored in cloud environments. This Equation is used to verify whether a provided point lies on a particular elliptic curve in  $F_p$ , which is significant for secure key validation. For example, the point  $\{x, y\}$  belongs to the curve in Eq. 3:

$$y^2 = x^3 + 7(\text{mod } 17) \quad (3)$$

This ensures that all cryptographic operations remain within the secure structure of the selected curve.

### Merkle Tree for Data Integrity Verification

The integration of a Merkle tree-based data integrity mechanism demonstrates an efficient application of the cybersecurity trust model. The Merkle tree validates data efficiency and tamper resistance, securing against unauthorized modifications during both transmission and storage. Fig. 2 presents the workflow of the proposed DT-ECC framework for secure cloud data encryption, decryption, and integrity verification.

A key advantage of the Merkle tree is that only a small subset of data is required to verify the integrity of an entire file, and this efficiency is independent of file size. In this scenario, the public portion of the Merkle tree is uploaded to the P2P Cloud Storage Server (CSS), and the user retains the remaining portion securely. Before verifying data integrity, the P2P CSS reacts to random client queries.

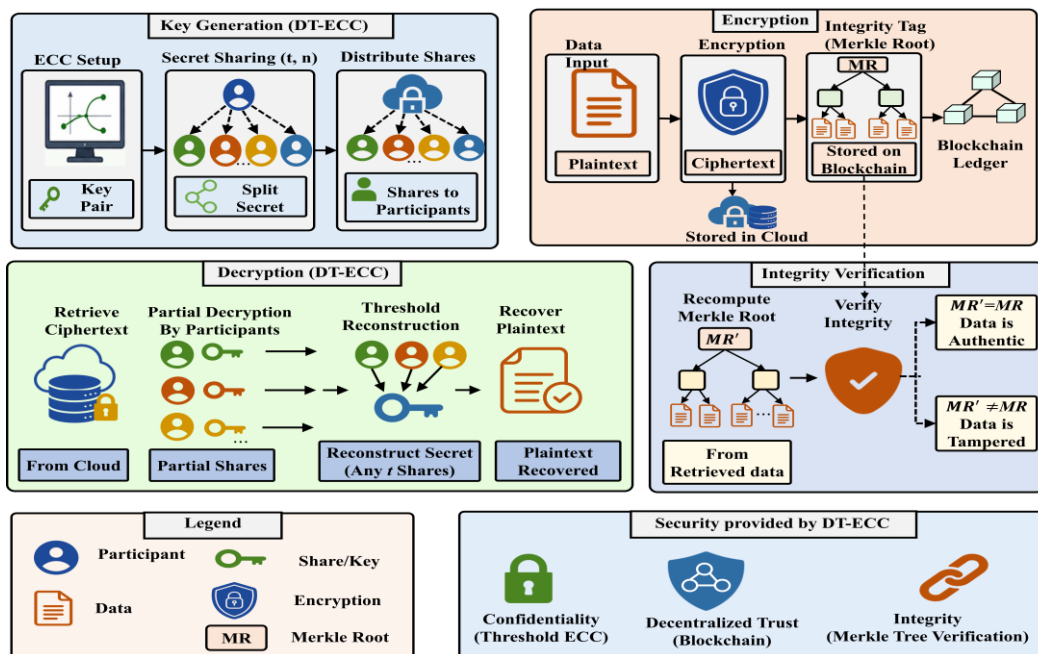


Fig. 2: Workflow of the proposed DT-ECC framework for secure cloud data encryption, decryption and integrity verification

Data are uploaded in chunks, which are arranged into the Merkle tree with branching patterns. This study also examines how variations in tree structure affect communication and computation overhead. Since the public portion of the Merkle tree must be sent to cloud servers, communication costs are directly proportional to its size. The Merkle trees are used for integrity verification; in the proposed model, they are applied to verify DT-ECC-encrypted cloud data while the root hash is immutably stored on the blockchain. Therefore, the novelty lies in the integrity verification with decentralized encryption and blockchain auditability.

#### Algorithm 1: Proposed DT-ECC-Based Secure Cloud Data Storage

Input: Cloud data ( $D$ ), threshold parameters  $((t, n))$ , participating nodes ( $N$ )

Output: Secure ciphertext ( $C$ ), verified recovered data ( $D'$ )

1. Start
2. Split input data ( $D$ ) into data segments  $(\{D_1, D_2, \dots, D_m\})$ .
3. Compute hash  $(H(D_i))$  for each data segment and construct the Merkle tree.
4. Generate the Merkle root ( $MR$ ).
5. Record  $MR$  and the corresponding transaction information on the permissioned blockchain.
6. Establish a secure session key  $MR$  between communicating entities using QKD.
7. Initialize ECC parameters and generate the distributed threshold key shares among  $n$  authorized nodes.
8. Verify that at least  $t$  authorized nodes participate in the threshold operation.
9. Generate the corresponding public key ( $PK$ ) from the distributed key shares.
10. Encrypt the data using the generated DT-ECC public key and obtain  $C$ .
11. Store  $C$  in the cloud and maintain the associated integrity information on the blockchain.
12. For data retrieval: obtain  $C$  from cloud storage and collect at least  $t$  valid key shares.
13. Perform partial decryption at the participating nodes.
14. Reconstruct the plaintext ( $D'$ ) from the valid partial decryptions.
15. Recalculate the Merkle root ( $MR'$ ) from the recovered data.
16. Compare ( $MR'$ ) with the blockchain-recorded root ( $MR$ ).
17. If ( $MR' = MR$ ): accept ( $D'$ ) as authentic and untampered.
18. Else: reject ( $D'$ ) and report an integrity violation.
19. End

## Results and Discussion

The DT-ECC is simulated in Python with a system configuration of Windows 10 OS, an Intel i5 processor, and 8GB RAM. The performance metrics include key generation time, response time, encryption time, decryption time, memory utilization, and latency. Data sizes of 10 MB, 15 MB, 20 MB, 25 MB, and 30 MB are considered for evaluation. The latency and memory utilization increase with data size but remain lower than

the evaluated baseline methods. The DT-ECC utilizes a fixed  $(t, n) = (3, 5)$  threshold configuration where  $n = 5$  represents the total number of authorized cloud nodes and  $t = 3$  is the minimum number of valid shares required for key reconstruction and threshold decryption for reproducibility. Each received share is verified independently against its commitment before it is accepted, where the invalid shares are rejected and not considered for threshold. DT-ECC is evaluated in a simulated permissioned blockchain environment with participating cloud nodes operating under a P2P communication model. All the experiments are conducted under the same network and software conditions to ensure consistency in performance evaluation. The cryptographic operations, threshold share generation, integrity verification, and blockchain transaction simulation are executed in a controlled software environment. The performance enhancements are mainly attributed to the distributed operation of DT-ECC, while the integrated QKD and Merkle tree improve security with limited additional overhead. QKD enhances the key establishment phase by preventing interference during secure key exchange, whereas the Merkle tree provides lightweight integrity validation of encrypted cloud data without the need for entire file reprocessing. Blockchain supports overall integration by preserving immutable logs of key commitments and integrity proofs.

Figs. 3 and 4 present the key generation time and response time, respectively, for DT-ECC compared with the Digital Signature Algorithm (DSA), RSA, Advanced Encryption Standard (AES), and ECC. DT-ECC achieves key generation times of 0.25 s, 1.25 s, 2.80 s, 4.10 s, and 5.65 s at 10 MB, 15 MB, 20 MB, 25 MB, and 30 MB, respectively. For the same data sizes, DT-ECC attains response times of 0.32 s, 1.55 s, 3.25 s, 4.85 s, and 6.55 s, respectively.

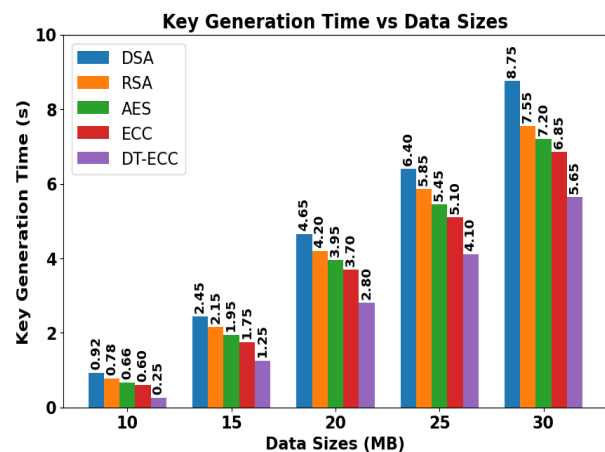


Fig. 3: Key generation time (s) analysis of the proposed DT-ECC framework and existing cryptographic methods across different data sizes

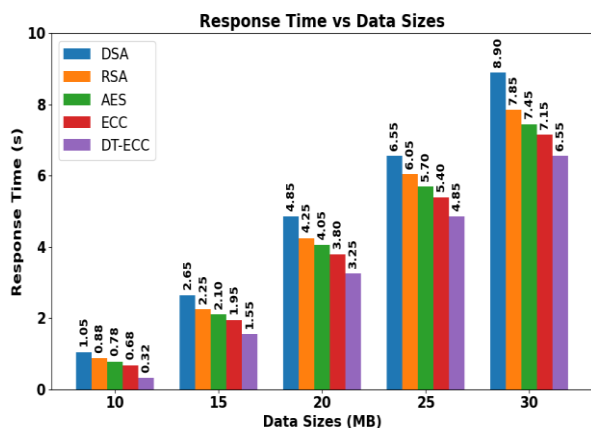


**Fig. 4:** Response time (s) analysis of the proposed DT-ECC framework and existing cryptographic methods across different data sizes

This improvement is attributed to distributed share computation, where portions of the key material are generated in parallel and then combined, thereby reducing the computational cost of centralized approaches such as RSA or DSA. Furthermore, DT-ECC utilizes lightweight threshold testing in response operation, avoiding redundant re-execution of the entire cryptographic procedure. This reduces initialization and overhead delays, thereby making DT-ECC applicable in dynamic applications.

Figs. 5 and 6 present the encryption and decryption times, respectively, for DT-ECC compared with DSA, RSA, AES, and ECC across the same data sizes. DT-ECC achieves encryption times of 0.10 s, 1.40 s, 3.05 s, 3.85 s, and 5.90 s, and decryption times of 0.07 s, 0.75 s, 3.00 s, 3.70 s, and 5.50 s, respectively. Unlike RSA and DSA, which rely on repeated modular exponentiations, DT-ECC distributes elliptic-curve scalar multiplications across multiple computers. Parallel partial scalars accelerate encryption, while aggregated partial shares expedite decryption, eliminating the need for complete standalone inversions. This distribution of cryptographic workloads lowers the per-operation complexity and enables faster, secure communication without compromising security strength.

Figs. 7 and 8 present the memory utilization and latency, respectively, for DT-ECC compared with DSA, RSA, AES, and ECC. DT-ECC records memory utilization of 35 MB, 110 MB, 255 MB, 375 MB, and 525 MB at 10 MB, 15 MB, 20 MB, 25 MB, and 30 MB, respectively. For the same data sizes, DT-ECC achieves latency values of 0.40 s, 1.65 s, 3.50 s, 5.05 s, and 6.70 s, respectively. These results highlight the advantage of compact distributed fragments of keys instead of storing large centralized key materials. Latency reduction is achieved through overlapping communication with partial computations executed concurrently, thereby avoiding sequential processing bottlenecks. This framework

enables faster end-to-end operations with minimized resource demand, thereby making DT-ECC effective with limited resources.

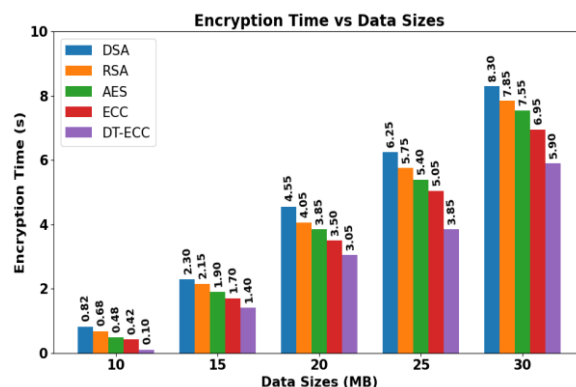


**Fig. 5:** Encryption time (s) analysis of the proposed DT-ECC framework and existing cryptographic methods across different data sizes

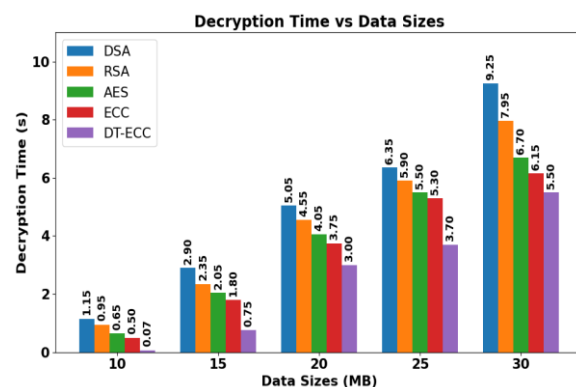


**Fig. 6:** Decryption time (s) analysis of the proposed DT-ECC framework and existing cryptographic methods across different data sizes

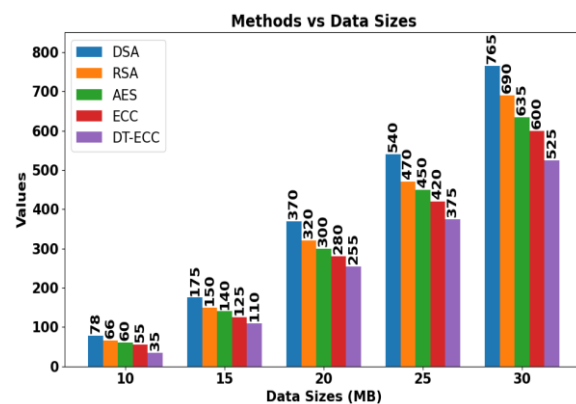


**Fig. 7:** Memory utilization (MB) analysis of the proposed DT-ECC framework and existing cryptographic methods across different data sizes

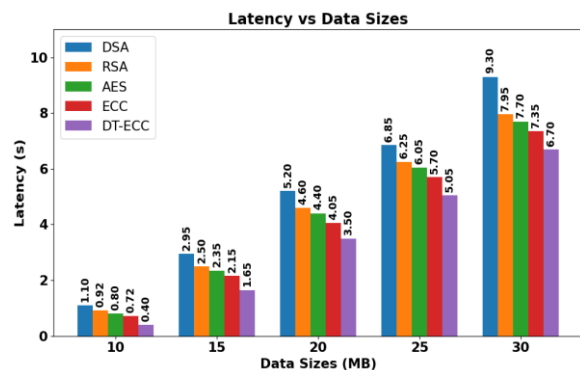


**Fig. 8:** Latency (s) analysis of the proposed DT-ECC framework and existing cryptographic methods across different data sizes

### Comparative Analysis

A comparative analysis of DT-ECC with three existing approaches, namely CR-FH-CPABE (Bai et al., 2024), MAES (Akbar et al., 2024), and CP-ABE (Tang et al., 2024), using a 30 MB dataset is presented in Table 1. The evaluation considered key generation time, response time, encryption time, decryption time, memory utilization, and latency. All methods are evaluated using 30 MB input data under the same simulation environment and software configuration. The parameter settings of existing methods are adapted from the original descriptions to ensure fair performance comparison. The results demonstrated that DT-ECC outperforms existing models, requiring less encryption time, key generation time, and memory utilization. The findings ensure that DT-ECC has higher efficiency and a lightweight nature, thereby making it suitable for cloud data storage environments where a high level of security is essential. The comparison results are

**Table 1:** Comparative performance evaluation of the proposed DT-ECC and existing methods for secure cloud data storage using 30 MB of data

| Methods                        | Key generation time (s) | Response time (s) | Encryption time (s) | Decryption time (s) | Memory utilization (MB) | Latency (s) |
|--------------------------------|-------------------------|-------------------|---------------------|---------------------|-------------------------|-------------|
| CR-FH-CPABE (Bai et al., 2024) | 7.85                    | 8.60              | 7.40                | 7.25                | 680                     | 9.85        |
| MAES (Akbar et al., 2024)      | 6.90                    | 7.55              | 6.80                | 6.65                | 610                     | 8.75        |
| CP-ABE (Tang et al., 2024)     | 6.45                    | 7.20              | 6.20                | 6.10                | 575                     | 7.95        |
| Proposed DT-ECC                | 5.65                    | 6.55              | 5.90                | 5.50                | 525                     | 6.70        |

These findings ensure that DT-ECC not only secures sensitive cloud data but also ensures operational efficiency by balancing high security with system performance. Overall, DT-ECC provides a better, flexible, and scalable approach to secure cloud data storage and transmission. DT-ECC enhances confidentiality and resilience by distributing key control, blockchain improves trust and traceability, QKD enhances key establishment security, and Merkle tree ensures effective integrity verification. This makes the framework more suitable for a real-time cloud storage environment than existing cryptographic models. DT-

based on the simulation-based performance under the same conditions.

### Scalability and Computational Overhead

The DT-ECC is estimated in terms of scalability and computational overhead, where scalability is analyzed based on the data size. As the data size increased from 10 MB to 30 MB, the framework shows an increase in execution time while maintaining latency and memory utilization, thereby showing better scalability. The distributed threshold key generation minimizes the computational burden on individual nodes for cryptographic operations. These results denote that DT-ECC is computationally efficient and suitable for scalable cloud storage environments.

### Discussion

DT-ECC addresses the limitations of conventional cryptographic models by decentralizing key generation and mitigating single points of failure. Unlike centralized systems, the threshold approaches distribute cryptographic functionality across multiple nodes, improving resistance to both internal and external attacks. Blockchain ensures auditability and transparency of key management, while the Merkle tree structure ensures tamper-proof data integrity. The performance analysis demonstrates improvements in key generation, encryption, decryption, and response times compared with RSA, DSA, and AES, thereby establishing DT-ECC as a lightweight architecture. Reduced memory usage and latency further support its applicability to resource-limited and real-time cloud infrastructure. Significantly, DT-ECC leverages parallel partial computations, reducing overhead and enabling scalability by increasing data size.

ECC enhances resilience by ensuring that the private key is not fully stored at a single node and by using blockchain-recorded commitments to detect inconsistent share behavior. Byzantine and adversarial simulations have not been conducted in the present study and will be considered in future work

### Conclusion

This study proposed DT-ECC integrated with blockchain and QKD to enhance cloud data security. DT-ECC successfully overcame the limitations of traditional

cryptographic approaches by decentralizing key generation and mitigating a single point of failure, thereby ensuring robust security of sensitive data. Blockchain enhanced transparency, tamper resistance, and auditability, while Merkle tree-based verification ensured integrity in distributed environments. The integration of QKD provided quantum-resilient key exchanges and protection against tampering from threats in cloud computing. Experimental analysis ensures that DT-ECC achieved better performance compared with existing models in terms of key generation time, response time, encryption time, decryption time, memory utilization, and latency. These results validated the lightweight and scalable nature of DT-ECC, thereby making it suitable for dynamic and resource-constrained cloud environments. By combining efficiency with high-level security, DT-ECC addressed key challenges in cloud computing and provided a resilient mechanism for secure data storage and transmission.

**Limitations and Future Work:** The proposed DT-ECC is evaluated in a simulation environment with QKD considered as a key establishment mechanism. Future work will focus on QKD deployment, large-scale cloud validation, optimization of the blockchain mechanism, and integration of post-quantum cryptographic techniques.

## Acknowledgment

The authors would like to express their sincere gratitude to the Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad, India, for providing the facilities, guidance, and support that enabled the successful completion of this research work.

## Funding Information

This research received no external funding.

## Author's Contributions

**Naga Swetha Dhulipudi:** Conceptualization, software, validation, formal analysis, data curation, write original draft preparation, visualization, funding acquisition.

**Balaji Savadam:** Conceptualization, methodology, investigation, resources, write review and edited, supervision, project administration, DNS.

## Ethics

This research does not involve any studies with human participants or animals, nor does it include any sensitive data, personal information, or experiments requiring ethical approval. Therefore, no ethical issues are applicable to this work.

## References

- Ahmad, S., & Mehfuz, S. (2024). Efficient time-oriented latency-based secure data encryption for cloud storage. *Cyber Security and Applications*, 2, 100027. <https://doi.org/10.1016/j.csa.2023.100027>
- Akbar, M., Waseem, M. M., Mehanoor, S. H., & Barmavatu, P. (2024). Blockchain-based cybersecurity trust model with multi-risk protection scheme for secure data transmission in cloud computing. *Cluster Computing*, 27(7), 9091–9105. <https://doi.org/10.1007/s10586-024-04481-9>
- Altahat, M. A., Daradkeh, T., & Agarwal, A. (2024). Optimized Encryption-Integrated Strategy for Containers Scheduling and Secure Migration in Multi-Cloud Data Centers. *IEEE Access*, 12, 51330–51345. <https://doi.org/10.1109/access.2024.3386169>
- Bai, Y., Fan, K., Zhang, K., Li, H., & Yang, Y. (2024). CR-FH-CPABE: Secure File Hierarchy Attribute-Based Encryption Scheme Supporting User Collusion Resistance in Cloud Computing. *IEEE Internet of Things Journal*, 11(10), 17727–17739. <https://doi.org/10.1109/jiot.2024.3358745>
- Bhattacharjee, S., Sharma, H., Choudhury, T., & Abdelmoniem, A. M. (2024). Leveraging chaos for enhancing encryption and compression in large cloud data transfers. *The Journal of Supercomputing*, 80(9), 11923–11957. <https://doi.org/10.1007/s11227-024-05906-3>
- Chowdhury, K., Deb, S., Kar, N., Sarkar, J. L., Alkhayyat, A. H., & Nkenyereye, L. (2025). PwLM $\mu$ -TPE: A Visual-Usability and Privacy-Enhanced Thumbnail Preserving Encryption Scheme of Cloud Computing for Consumers. *IEEE Transactions on Consumer Electronics*, 71(1), 9–17. <https://doi.org/10.1109/tce.2024.3412100>
- Dhamdhere, S. D., Sivakkumar, M., & Subramanian, V. (2025). Cloud data security with deep maxout assisted data sanitization and restoration process. *High-Confidence Computing*, 5(1), 100238. <https://doi.org/10.1016/j.hcc.2024.100238>
- Gupta, A., Namasudra, S., & Kumar, P. (2024). A secure VM live migration technique in a cloud computing environment using blowfish and blockchain technology. *The Journal of Supercomputing*, 80(19), 27370–27393. <https://doi.org/10.1007/s11227-024-06461-7>
- Jin, W. (2024). RETRACTED ARTICLE: Security and privacy of digital economic risk assessment system based on cloud computing and blockchain. *Soft Computing*, 28(3), 2753–2768. <https://doi.org/10.1007/s00500-023-09586-8>

- Kathole, A. B., Vhatkar, K. N., Goyal, A., Kaushik, S., Mirge, A. S., Jain, P., Soliman, M. S., & Islam, M. T. (2024). Secure Federated Cloud Storage Protection Strategy Using Hybrid Heuristic Attribute-Based Encryption with Permissioned Blockchain. *IEEE Access*, 12, 117154–117169. <https://doi.org/10.1109/access.2024.3447829>
- Krishnan, S. C. (2024). AI-HybridChain: Picturized authentication and DRL based access control method with secure two fold revocation for ensuring cloud computing security. *Future Generation Computer Systems*, 160, 389–405. <https://doi.org/10.1016/j.future.2024.04.054>
- Li, J., Zhang, E., Han, J., Zhang, Y., & Shen, J. (2025). PH-MG-ABE: A Flexible Policy-Hidden Multigroup Attribute-Based Encryption Scheme for Secure Cloud Storage. *IEEE Internet of Things Journal*, 12(2), 2146–2157. <https://doi.org/10.1109/jiot.2024.3468018>
- Li, W., Susilo, W., Xia, C., Huang, L., Guo, F., & Wang, T. (2024). Secure Data Integrity Check Based on Verified Public Key Encryption With Equality Test for Multi-Cloud Storage. *IEEE Transactions on Dependable and Secure Computing*, 21(6), 5359–5373. <https://doi.org/10.1109/tdsc.2024.3375369>
- Liu, Q., Zhou, F., & Chen, H. (2024). Secure medical data on cloud storage via DNA homomorphic encryption technique. *Physical Communication*, 64, 102295. <https://doi.org/10.1016/j.phycom.2024.102295>
- Mondal, A., & Chatterjee, P. S. (2024). CloudSec: A Lightweight and Agile Approach to Secure Medical Image Transmission in the Cloud Computing Environment. *SN Computer Science*, 5(2), 237. <https://doi.org/10.1007/s42979-023-02539-w>
- Ojha, S., Paygude, P., Dhumane, A., Rathi, S., Bidve, V., Kumar, A., & Devale, P. (2024). A method to enhance privacy preservation in cloud storage through a three-layer scheme for computational intelligence in fog computing. *MethodsX*, 13, 103053. <https://doi.org/10.1016/j.mex.2024.103053>
- Pedersen, T. P. (2007). Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. *Advances in Cryptology — CRYPTO '91*, 129–140. [https://doi.org/10.1007/3-540-46766-1\\_9](https://doi.org/10.1007/3-540-46766-1_9)
- Pradhan, G., & Priyadarsini, M. (2024). A trusted computing framework for cloud data security using role-based access and pattern recognition. *Cluster Computing*, 27(5), 6609–6622. <https://doi.org/10.1007/s10586-024-04274-0>
- Shakor, M. Y., Khaleel, M. I., Safran, M., Alfarhood, S., & Zhu, M. (2024). Dynamic AES Encryption and Blockchain Key Management: A Novel Solution for Cloud Data Security. *IEEE Access*, 12, 26334–26343. <https://doi.org/10.1109/access.2024.3351119>
- Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11), 612–613. <https://doi.org/10.1145/359168.359176>
- Shrivastava, P., Alam, B., & Alam, M. (2024). A hybrid lightweight blockchain based encryption scheme for security enhancement in cloud computing. *Multimedia Tools and Applications*, 83(1), 2683–2702. <https://doi.org/10.1007/s11042-023-17040-y>
- Sultana, N. M., & Srinivas, K. (2024). Data Privacy Protection in Cloud Computing Using Visual Cryptography. *Multimedia Tools and Applications*, 84(21), 23577–23597. <https://doi.org/10.1007/s11042-024-19963-6>
- Tang, X., Guo, C., Choo, K.-K. R., Jiang, X., & Liu, Y. (2024). A secure and lightweight cloud data deduplication scheme with efficient access control and key management. *Computer Communications*, 222, 209–219. <https://doi.org/10.1016/j.comcom.2024.05.003>
- Umar, T., Nadeem, M., & Anwer, F. (2024). Chaos based image encryption scheme to secure sensitive multimedia content in cloud storage. *Expert Systems with Applications*, 257, 125050. <https://doi.org/10.1016/j.eswa.2024.125050>
- Yihui, Z. (2024). Design of cloud data storage security and financial risk control management early warning system based on sensor networks. *Measurement: Sensors*, 32, 101064. <https://doi.org/10.1016/j.measen.2024.101064>
- Zhang, Z., Zhou, N., Sun, B., Banerjee, S., & Mou, J. (2024). Multimedia healthcare cloud personal archives security system based on compressed sensing and multi-image encryption. *Journal of the Franklin Institute*, 361(8), 106844. <https://doi.org/10.1016/j.jfranklin.2024.106844>
- Zubair, S., & Ahmed, H. M. A. (2024). A hybrid algorithm-based optimization protocol to ensure data security in the cloud. *International Journal of Information Technology*, 16(5), 3057–3064. <https://doi.org/10.1007/s41870-023-01546-7>