Received: 27 December 2024 | Revised: 20 March 2025 | Accepted: 3 April 2025 | Published online: 28 May 2025

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2025, Vol. 00(00) 1–19
DOI: 10.47852/bonviewJCCE52025101

BON VIEW PUBLISHING

Decentralized Big Data Auditing Scheme for Cloud Storage Based on Blockchain with Adaptive EI-GAMAL and Gazelle Optimization

Showri Rayalu Bandanadam[1] and Prasanna Kumar Rangarajan[1,*]

[1]*Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, India*

Abstract: Cloud service providers may accidentally damage or delete user data in cloud storage, leading to data loss without user notification. To mitigate this, public auditing mechanisms are becoming increasingly crucial. However, many existing systems rely on third-party auditors (TPAs), which provide efficiency and fairness but remain vulnerable to malicious behavior. This risk stems from reliance on a centralized third party, highlighting the need for more secure methods. Blockchain technology offers a viable solution to mitigate this risk. By decentralizing the auditing process, blockchain eliminates reliance on a TPA, ensuring that data integrity validation is distributed and secure. Blockchain's transparency and immutability make it ideal for strengthening data auditing in cloud storage. In this enhanced auditing approach, cloud providers collaborate to validate data, establishing a decentralized framework. The process begins with the collection of data from traditional databases and its division into blocks for encryption. The adaptive EI-GAMAL algorithm, enhanced by the Enhanced Predator Success Rate of Gazelle Optimization, encrypts the data. The encrypted blocks are then stored in the cloud using the divide and conquer table (D&CT) concept, ensuring continuous updates to the location and metadata associated with the data. Each block contains a file ID, user ID, file data, and version number, which updates upon data modification or deletion. The location table keeps track of the file's location, which is also updated during the D&CT operation. This mechanism safeguards sensitive data and ensures its integrity through decentralized auditing. The performance of this blockchain-based auditing approach is validated against traditional methods, demonstrating greater effectiveness and security.

Keywords: decentralized big data auditing, blockchain, key optimization, divide and conquer table, adaptive EI-GAMAL, enhanced predator success rate of gazelle optimization

1. Introduction

Cloud computing has gained significant popularity among individual users and businesses due to its ability to provide services as general utilities, such as water and electricity [1]. It serves as an efficient mechanism for managing and delivering information and communication technology resources to remote users [2]. By leveraging virtualization techniques, cloud computing enables the efficient utilization of a vast pool of connected computing assets, including services, computing power, and storage [3]. As a rapidly emerging organizational computing approach, it offers several advantages, such as scalability, cost efficiency, and ample storage capacity [4]. Many organizations choose to outsource their data to cloud servers for processing and storage [5]. However, ensuring data integrity and security in cloud environments is a critical challenge, particularly given the increasing reliance on big data analytics [6].

Verifying the integrity of data stored on cloud servers is a complex and urgent challenge in cloud computing [7]. Data auditing mechanisms allow users to ensure the integrity of their data on remote cloud servers without needing to retrieve it [8]. Based on the role of the verifier, data auditing approaches can be classified into public and private auditing [9]. Enhancing blockchain methodologies presents a promising solution by offering an append-only distributed database model that supports the development of a decentralized, transparent, and secure platform for cloud storage [10]. A blockchain consists of multiple interconnected nodes managed through distributed ledger technology, making it highly resistant to tampering and forgery [11]. Once data is recorded on the blockchain, it cannot be modified, ensuring strong data integrity [12]. This feature makes blockchain a more secure alternative to conventional data verification methods. However, blockchain technology cannot be directly integrated into traditional auditing mechanisms due to its inefficiency in handling large volumes of data [13]. Additionally, incorporating third-party auditors (TPAs) into a decentralized blockchain system is challenging due to the centralized nature of TPAs [14].

***Corresponding author:** Prasanna Kumar Rangarajan, Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, India. Email: r_prasannakumar@ch.amrita.edu

Significant security challenges hinder the effectiveness of conventional cloud-based big data auditing mechanisms [15]. One of the primary concerns is ensuring data integrity for users operating on untrusted platforms provided by cloud service providers (CSPs) [16]. Since outsourced data is stored remotely, users lose direct control over their information, raising concerns about data security and ownership [17]. Moreover, the integration of data in cloud storage is closely linked to veracity, one of the four key characteristics of big data—velocity, volume, veracity, and variety [18]. Veracity, which refers to the accuracy and reliability of data, is difficult to maintain in cloud environments [19]. Furthermore, securing large volumes of information in cloud storage presents additional security risks [20]. To address these challenges, this study explores the integration of blockchain and data auditing mechanisms, leading to the development of a blockchain-based data auditing framework for cloud storage. The key objectives of the proposed decentralized big data auditing strategy for cloud storage based on blockchain are as follows:

1) To design a decentralized big data auditing mechanism that enhances the reliability and integrity of the auditing process.
2) To encrypt data using the adaptive EI-GAMAL (AEI-GAMAL) encryption scheme, with key optimization achieved through the Enhanced Predator Success Rate of Gazelle Optimization (EPSRGO) algorithm.
3) To develop the EPSRGO algorithm by improving the traditional Gazelle Optimization Algorithm (GOA) for key optimization, thereby enhancing performance.
4) To store encrypted data in the cloud using the divide and conquer table (D&CT) approach, which updates the information table and location array to facilitate the data auditing process.
5) To evaluate the proposed decentralized big data auditing mechanism by comparing it with traditional optimization algorithms and cryptographic approaches across various performance metrics.

The decentralized big data auditing strategy for cloud storage based on blockchain includes the upcoming sections. Section 2 provides an overview of existing data auditing mechanisms. Section 3 presents the decentralized framework for big data auditing in cloud storage, focusing on adaptive encryption and enhanced optimization techniques. Section 4 details the AEI-GAMAL encryption approach and the enhanced optimization strategy for secure cloud storage over a blockchain network. Section 5 introduces an intelligent data auditing scheme for big data in the cloud using the D&CT model. Section 6 discusses the solutions offered by the proposed decentralized big data auditing mechanism based on blockchain. Finally, Section 7 concludes the study by summarizing key findings and future research directions.

2. Existing Works

2.1. Related works

In 2020, Mohan et al. [21] presented a cloud data auditing method using Merkle trees and blockchain. Merkle trees enable efficient data integrity verification, while blockchain ensures secure, immutable audit logs. The proposed model improves security and reduces computational overhead compared to traditional methods.

In 2019, Yu et al. [22] proposed a decentralized big data auditing framework for smart city environments using blockchain technology. By leveraging smart contracts, the system ensures secure, transparent, and automated data auditing without relying on third-party auditors (TPAs). The approach enhances data integrity and privacy but faces challenges related to blockchain scalability and computational overhead.

In 2025, Liu et al. [23] presented a blockchain-assisted framework for fine-grained data deduplication and integrity auditing in cloud storage. By combining deduplication techniques with blockchain technology, the model enhances storage efficiency while ensuring data integrity and security. The approach effectively reduces redundancy and computational overhead but faces challenges related to blockchain scalability and transaction delays.

In 2015, Liu et al. [24] proposed MuR-DPA, a secure public auditing framework for dynamic big data storage in cloud environments. By using a multi-replica Merkle hash tree (MHT) with a top-down leveled structure, the model efficiently verifies data integrity while supporting dynamic data operations such as updates and deletions. The approach enhances audit efficiency but introduces computational overhead in managing multiple data replicas.

In 2020, Lekshmi et al. [25] proposed a blockchain-based data auditing system using smart contracts to automate and secure the audit process in cloud storage. The approach enhances data integrity, reduces reliance on TPAs, and ensures transparent, tamper-proof audit logs. However, challenges like smart contract vulnerabilities and deployment costs are identified.

In 2022, Shu et al. [26] proposed a blockchain-based decentralized public auditing framework for cloud storage. By integrating smart contracts, the system automates the audit process, ensuring secure, transparent, and tamper-proof data verification without relying on TPAs. While enhancing trust and security, the method faces challenges such as blockchain scalability and transaction costs.

In 2019, Fan et al. [27] presented Dredas, a blockchain-based decentralized data auditing scheme designed for Industrial IoT environments. By leveraging smart contracts, the system automates and enhances the reliability of data auditing while ensuring security and transparency. The approach efficiently handles large-scale IoT data but may face challenges such as network latency and smart contract vulnerabilities.

In 2020, Li et al. [28] proposed a blockchain-based public auditing framework for big data in cloud storage. The system ensures secure, transparent, and tamper-proof audit logs while enabling TPAs to verify data integrity without compromising user privacy. Although effective for large-scale data environments, it faces challenges related to blockchain scalability and storage overhead.

In 2025, Zhang et al. [29] presented a blockchain-based framework for privacy-preserving deduplication and integrity auditing in cloud storage. The proposed system ensures data privacy and integrity by leveraging blockchain's decentralized and immutable features. It incorporates privacy-preserving techniques to deduplicate data efficiently while enabling secure auditing, reducing the risks associated with unauthorized access or data tampering. The approach addresses challenges such as maintaining data confidentiality, scalability, and performance in cloud environments.

2.2. Research gaps and challenges

The concept of big data has gained significant attention from both academic and governmental sectors worldwide. Big data is generated from various sensor networks and computational technologies and is stored in cloud environments. Ensuring the auditability and integrity of big data is essential for maintaining its analytical functionality. The cloud environment offers flexible services, allowing data owners to access high-performance computing resources anytime and anywhere. Additionally, data auditing enables data owners to detect malicious activities by CSPs that may compromise data integrity. Numerous studies have been conducted to develop decentralized big data auditing schemes for cloud storage, addressing key merits and limitations, as illustrated in Table 1. Blockchain technology [30] minimizes cost utilization and enhances data security. However, it consumes a significant amount of power and faces scalability limitations due to its restricted block size. The MHT [31] is widely used to verify data integrity with minimal disk space requirements. However, it has high time complexity and security vulnerabilities. Blockchain technology [32] also improves transparency and trust, ensuring data fidelity. Yet, it is prone to private key management issues and makes data modification difficult. The fair trade mechanism [24] enhances data security and ensures fair transactions while enabling asset tokenization. However, it is costly and time-consuming, making it unsuitable for real-time applications. Blockchain [33] effectively prevents

Table 1
Features and challenges of conventional decentralized big data auditing scheme for cloud storage

Author [citation]	Methodology	Features	Challenges
Mohan et al. [21]	Utilizes Merkle trees for data integrity verification and blockchain for secure, immutable audit logs.	• Efficient data integrity checks • Enhanced security through blockchain immutability • Reduced computational overhead	• Potential latency in blockchain transactions • Storage overhead due to blockchain data growth
Yu et al. [22]	Blockchain-based decentralized auditing framework using smart contracts for big data in smart cities.	• Eliminates the need for third-party auditors • Ensures data integrity via blockchain immutability • Smart contracts automate auditing tasks	• Blockchain scalability issues in large-scale data environments • Potential computational overhead for smart contract execution
Liu et al. [23]	Blockchain-based framework combining fine-grained data deduplication and integrity auditing for cloud storage.	• Efficient data deduplication for storage optimization • Blockchain ensures secure and immutable audit logs • Reduces computational overhead during audits	• Potential scalability issues with blockchain growth • Latency concerns during blockchain transactions
Liu et al. [24]	MuR-DPA framework using a multi-replica Merkle hash tree for secure public auditing of dynamic big data in cloud storage.	• Efficient multi-replica auditing • Supports dynamic data operations (insertion, deletion, updates) • Reduces audit complexity with a top-down leveled structure	• Increased computation overhead for managing multiple data replicas • Potential scalability concerns for extremely large data sets

(*Continued*)

Table 1
(*Continued*)

Author [citation]	Methodology	Features	Challenges
Lekshmi et al. [25]	Blockchain-based auditing system using smart contracts for secure and automated data verification in cloud storage.	• Automated audit process through smart contracts • Ensures secure, tamper-proof audit logs • Reduces reliance on third-party auditors	• Potential vulnerabilities in smart contract code • Deployment costs associated with blockchain integration
Shu et al. [26]	Blockchain-based decentralized public auditing framework with smart contracts for cloud data verification.	• Eliminates reliance on third-party auditors • Ensures transparent, tamper-proof audit logs • Improves security against data tampering	• Blockchain scalability issues for large data volumes • Transaction costs may increase with frequent audits
Fan et al. [27]	Dredas: Blockchain-based decentralized data auditing scheme using smart contracts for Industrial IoT environments.	• Decentralized, reducing reliance on central auditors • Automated audit processes via smart contracts • Efficiently handles large-scale Industrial IoT data	• Potential network latency in large IoT systems • Smart contract vulnerabilities may pose security risks
Li et al. [28]	Blockchain-based public auditing framework for big data in cloud storage.	• Ensures secure, tamper-proof audit logs • Enables public auditing without compromising user privacy • Optimized for large-scale cloud data environments	• Blockchain scalability issues with large data volumes • Increased storage overhead due to audit log growth
Zhang et al. [29]	Blockchain-based framework for secure cloud storage, focusing on privacy-preserving deduplication and integrity auditing.	• Protects data confidentiality during deduplication and auditing • Efficiently eliminates redundant data, saving storage space	• Efficiency and performance in large datasets • Blockchain operations and cryptographic techniques can increase computational costs

third-party intervention in transactions. However, it encounters challenges in data updates and is complex to implement. The MHT [34] provides efficient data verification and prevents duplicate transactions. However, it has high CPU consumption and is computationally inefficient. Similarly, the MHT [35] can securely encrypt and fragment data blocks, ensuring data integrity. However, it is computationally intensive and difficult to implement in new systems. The integrity verification scheme [36] protects against malicious attacks and ensures data privacy. However, it fails to address data quality concerns and exhibits poor performance efficiency. To address these challenges, a novel decentralized big data auditing scheme for cloud storage based on blockchain technology is proposed. This approach aims to enhance data security, transparency, and auditability while mitigating the limitations of existing methodologies.

3. Decentralized Framework of Big Data Auditing Scheme for Cloud Storage: Adaptive Encryption and Enhanced Optimization

3.1 Description of proposed system

The increasing popularity of cloud computing and the rapid advancement of high-speed internet have significantly enhanced

collaborative work environments and data-sharing capabilities, particularly in the fields of artificial intelligence and big data. The evolution of cloud computing has led to substantial advancements in technology and computer science [37, 38]. As a result, many organizations are adopting cloud services, which allow data owners to access high-performance computing resources remotely at any time [39]. This capability reduces the reliance on local computing infrastructure, thereby minimizing costs associated with resource management. Additionally, cloud consumers are relieved from the burden of managing complex computing systems and local data storage [40].

Despite these advantages, cloud storage introduces several security challenges, including malicious modifications, data corruption, unauthorized access, and potential data loss [41]. As data volumes continue to grow in the big data sector, numerous research efforts have focused on improving security and integrity mechanisms. Ensuring data integrity is crucial for extracting accurate insights from big data while maintaining security [42]. Traditional data auditing mechanisms rely on the Merkle Hash Tree (MHT) to verify data integrity by comparing its root hash with the cloud-stored data [43]. However, these conventional approaches fail to provide robust security and integrity guarantees.

An enhanced blockchain-based mechanism offers a decentralized, transparent, and trust-enhancing framework for cloud storage security. However, blockchain technology cannot be directly integrated into traditional data auditing mechanisms due to its inefficiency in handling large-scale data records. The proposed decentralized big data auditing mechanism for cloud storage using blockchain technology is diagrammatically illustrated in Figure 1.

An enhanced data auditing strategy for cloud storage is developed based on blockchain technology to maximize the reliability and scalability of data auditing results. Unlike traditional methods, this approach eliminates the need for a TPA, as CSPs collaborate to verify data integrity, forming a decentralized blockchain network. Initially, data is collected from classical data sources and partitioned into blocks for encryption. The encryption process is performed using the AEI-GAMAL algorithm, optimized with the EPSRGO algorithm. Once encrypted, the data is securely stored in the cloud using the D&CT framework. Within this framework, both the data table and the data array are continuously updated to facilitate the data auditing process.

The data array includes key attributes such as the file ID, version number, file data, and user ID, which remain consistent across all blocks. The version number updates dynamically based on deletion or modification operations. Additionally, each file's region contains a location table that records its storage location, which is also updated whenever D&CT operations modify the data. This process ensures that sensitive data integrity is preserved within the cloud environment.

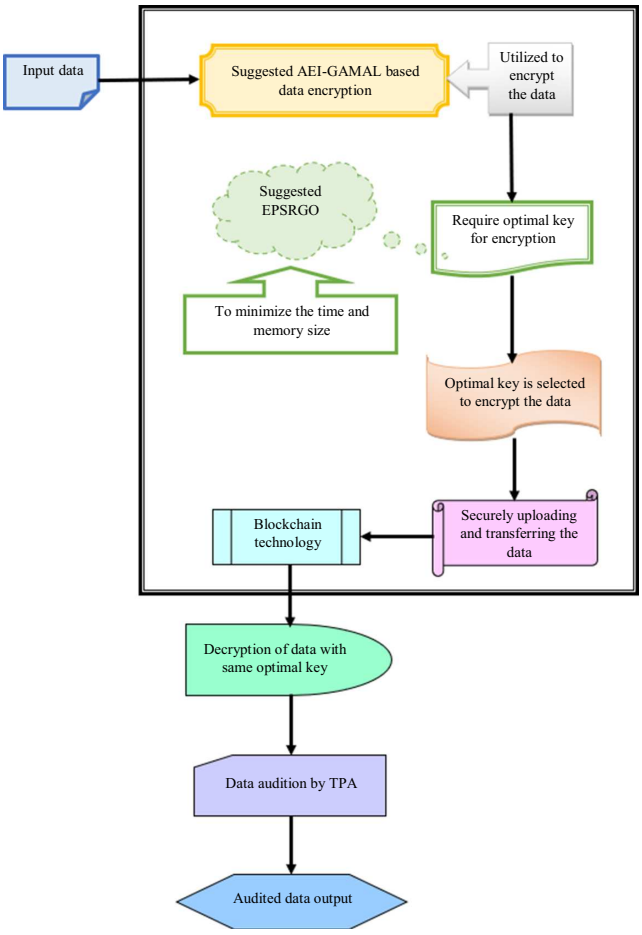
The performance of the proposed decentralized public auditing strategy for data storage is evaluated against conventional mechanisms across various performance metrics to assess its effectiveness and efficiency.

The decentralized public auditing strategy designed for data storage was validated against traditional mechanisms, using various performance factors to ensure its effectiveness, reliability, and efficiency.

3.1.1. Benchmarking against established methods

Traditional mechanisms, such as TPAs and cryptographic integrity checks, provide well-known performance benchmarks. Comparing the decentralized approach against these ensures that it meets or exceeds existing standards.

Figure 1
The diagrammatic representation of recommended decentralized big data auditing mechanism for cloud storage based on blockchain



3.1.2. Security and integrity assurance

Traditional auditing techniques rely on cryptographic proofs like message authentication codes, homomorphic authenticators, or Merkle trees. Evaluating the decentralized strategy against these methods ensures that it provides equivalent or improved data integrity verification.

3.1.3. Performance metrics evaluation

Computational Overhead: Ensuring that the decentralized auditing does not introduce excessive computation compared to centralized or traditional methods.

Storage Overhead: Comparing how much additional metadata or cryptographic proofs the new system requires.

Verification Time: Checking if the decentralized auditing process is efficient and does not introduce delays.

3.1.4. Scalability and latency analysis

Traditional approaches may face bottlenecks when dealing with large-scale cloud storage. Testing the decentralized method under similar conditions helps assess its scalability and latency in real-world applications.

3.1.5. Eliminating single points of failure

Traditional auditing often relies on a central entity (e.g., a trusted TPA). The decentralized approach must prove that it reduces central points of failure while maintaining or improving verification accuracy.

3.1.6. User privacy and trust

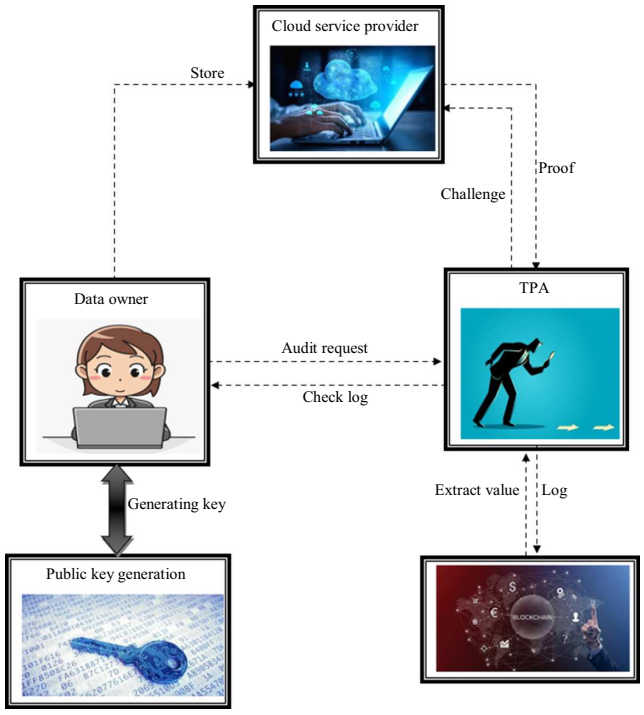
Some traditional methods require users to expose their data to TPAs, posing privacy concerns. The decentralized strategy should demonstrate that it provides equivalent or better privacy protection while maintaining transparency.

3.2. Auditing mechanism in cloud sector

The cloud auditing mechanism is used to generate hash values for encrypted data retrieved from cloud servers and to create digital signatures, ensuring data integrity verification. However, many existing mechanisms fail to protect user privacy, making data accessible to external auditors. Users do not require an auditing technique that introduces new data leakage risks, especially when external auditors are only responsible for validating data accuracy as per contractual agreements.

Additionally, auditing is resource-intensive, as data owners often have limited computational resources while managing large volumes of data. To address this challenge, the auditing process is delegated to multiple designated auditors responsible for verifying the accuracy of cloud-stored data. The cloud auditing mechanism is illustrated in Figure 2.

Figure 2
The illustration of auditing mechanism in the cloud sector



4. Adaptive EI-GAMAL for Data Encryption and Enhanced Optimization for Cloud Storage over the Blockchain Network

4.1 Enhanced predator success rate of gazelle optimization

The EPSRGO algorithm is an improved version of the traditional GOA, which is inspired by the behavioral characteristics of gazelles. The conventional GOA models both the grazing behavior of gazelles and their ability to evade predators. It is effective in solving real-time optimization problems and producing high-quality results.

However, in the traditional GOA, the predator success rate (PSR) is fixed at 0.34, limiting the model's adaptability and flexibility. To address this limitation, the EPSRGO algorithm dynamically estimates the PSR based on the fitness values of the GOA, enhancing its overall performance. The GOA formulation is provided in Equation (1).

$$PSR = \frac{mean\ fit}{wrst\ fit} * bst\ fit \tag{1}$$

Here, the worst and mean fitness measures of the GOA are specified. Next, the best fitness of the GOA is defined. The working process of the conventional GOA is given as follows:

The traditional GOA [44] is inspired by the survival capabilities of gazelles. Gazelles are among the primary food sources for various predators. They are social animals and have a strong instinct for escaping predators. Gazelles possess a keen sense of smell, sight, and hearing. These traits inspired the design of the GOA algorithm. The modeling of the GOA is presented in this section.

The first stage of the GOA is the initialization of the member population, as defined in Equation (2).

$$Q = \begin{bmatrix} q_{2,1} & q_{2,2} & \cdots & q_{2,e-1} & q_{2,e} \\ \vdots & \vdots & q_{j,k} & \vdots & \vdots \\ q_{m,1} & q_{m,2} & \cdots & q_{m,e-1} & q_{m,e} \end{bmatrix} \tag{2}$$

The present member population is denoted as Q. The variable $q_{j,k}$ points to the place of the j^{th} population in the k^{th} dimension, and the issue dimension is denoted as e. The overall member populations are specified as m.

The population of the GOA is created arbitrarily utilizing Equation (3).

$$q_{j,k} = rd \times (ub_k - lb_k) + lb_k \tag{3}$$

The factor rd denotes the arbitrary integer. The issue's upper and lower bounds are indicated as ub_k *and* lb_k correspondingly.

The fittest or strongest gazelles excel at evading predators, detecting threats, and alerting others. Thus, the optimal solution is represented as the top gazelle to generate an Elite matrix. The Elite matrix is used to determine and search for the gazelle's next position. The Elite matrix is formulated in Equation (4).

$$Elite = \begin{bmatrix} q'_{1,1} & q'_{1,2} & \cdots & q'_{1,e-1} & q'_{1,e} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,e-1} & q'_{2,e} \\ \vdots & \vdots & q'_{j,k} & \vdots & \vdots \\ q'_{m,1} & q'_{m,2} & \cdots & q'_{m,e-1} & q'_{m,e} \end{bmatrix} \tag{4}$$

The top gazelle vector is indicated as $q'_{j,k}$, and that is copied times to create the matrix.

The exploitation stage models gazelles grazing in the absence of predators. At this stage, gazelles move in Brownian motion. Equation (5) defines this phase.

$$\overrightarrow{gazelle}_{n+1} = \overrightarrow{gazelle}_n + t.\vec{S} * \vec{S}_C * \left(\overrightarrow{Elite}_n - \vec{S}_C * \overrightarrow{gazelle}_n\right) \tag{5}$$

Here, the next execution answer is denoted as $\overrightarrow{gazelle}_{n+1}$, and the present execution answer is indicated as $\overrightarrow{gazelle}_n$. The gazelle's grazing speed is pointed as t, and the term $\vec{S}_C$ refers to the vector that includes the arbitrary integer specifying the Brownian motion.

The uniform arbitrary integer's vector is referred to as S, and that falls in the interval between 0 and 1.

The exploitation phase is regarded as the escape stage from a predator. The gazelle begins running once it detects a predator. At this stage, the gazelle reacts by employing a Lévy flight. Equation (6) models the gazelle's behavior in this phase.

$$\overrightarrow{gazelle}_{n+1} = \overrightarrow{gazelle}_n + T.\alpha.CF * \vec{S} * \vec{S}_C * \left(\overrightarrow{Elite}_n - \vec{S}_N * \overrightarrow{gazelle}_n\right) \tag{6}$$

Here, the direction's sudden change is denoted as α, and the top speed is indicated as T. The factor $\vec{S}_N$ represents the vector of arbitrary integers according to the Levy distributions. The predator's behavior in this phase is formulated in Equation (7).

$$\overrightarrow{gazelle}_{n+1} = \overrightarrow{gazelle}_n + T.\alpha.CF * \vec{S}_C * \left(\overrightarrow{Elite}_n - \vec{S}_N * \overrightarrow{gazelle}_n\right) \tag{7}$$

Here, the term $CF = \left(1 - \frac{u}{U_{\max}}\right)^{\left(2\frac{u}{U_{\max}}\right)}$ indicates the attribute that manages the predator's movement.

The effect of PSR is estimated in Equation (8).

$$\overrightarrow{gazelle}_{n+1} = \begin{cases} \overrightarrow{gazelle}_n + CF\left[\vec{lb} + \vec{S} * \left(\vec{ub} - \vec{lb}\right)\right] * \vec{V} & if\ s \le PSRs \\ \overrightarrow{gazelle}_n + [PSRs\,(1-s) + s]\left(\overrightarrow{gazelle}_{s1} - \overrightarrow{gazelle}_{s2}\right) & else \end{cases} \tag{8}$$

In the conventional GOA, the PSR value is set to 0.34 that degrades the system's efficacy. So, the value of PSR is newly determined by using Equation (1) in the proposed approach.

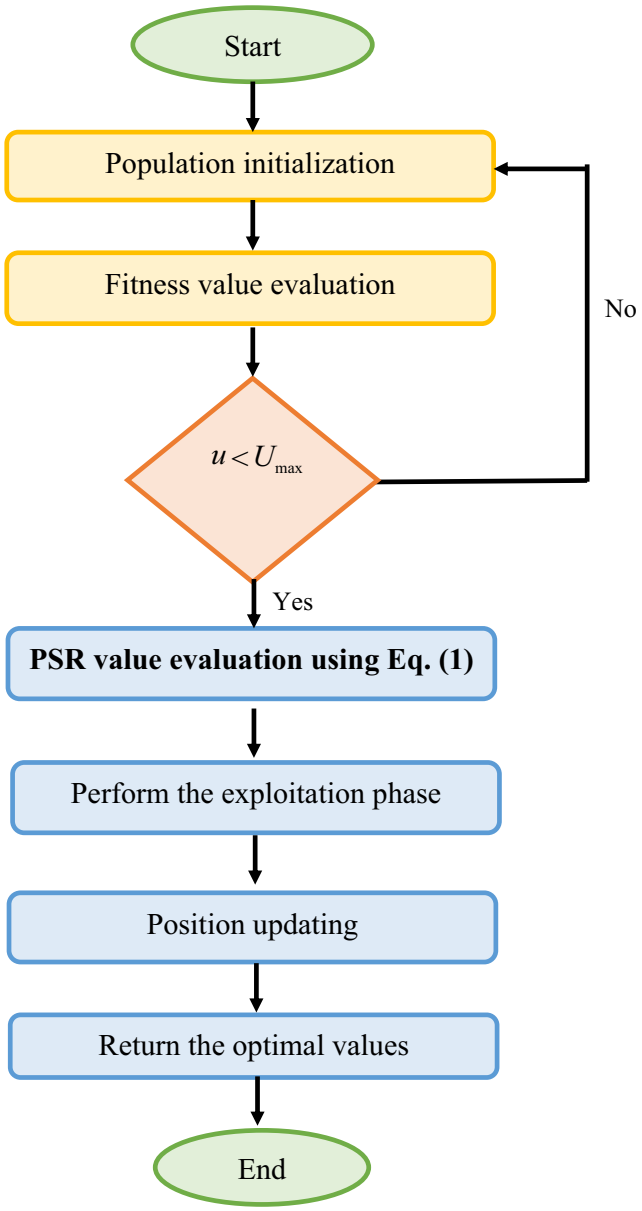
Here, the attribute $\vec{V}$ indicates the binary vector that is generated by creating an arbitrary integer s in [0, 1] such that $\vec{V} = \begin{cases} 0, & if\ s < 0.2 \\ 1, & otherwise \end{cases}$ s_1 *and* s_2 are the random indices of the matrix of gazelle. The flow chart of the recommended EPSRGO is offered in Figure 3. Moreover, Algorithm 1 illustrates the pseudo-code of the recommended EPSRGO task.

Algorithm 1

Implemented EPSRGO

The **population** variables and iteration count initialization.
The objective function calculation.
For $u = 1$ *to* $U_{\max}$
For $j = 1$ *to* N_{pop}
Estimate the PSR value utilizing Equation (1)**.**
Perform the Fitness-Oriented Assessment (FOA) scheme.
Execute the exploitation stage based on the Brownian motion and Levy flight in Equations (5) and (6).
Formulate the predator's behavior by applying Equation (7).
Model the PSR effect by adopting Equation (8).
Update the better positions.
End
End
Process the executions iteratively to attain optimal solutions.
Return the optimal answers.
End

Figure 3
The flow chart of the recommended EPSRGO



4.2. Conventional EI-GAMAL technique

The EI-GAMAL algorithm [45] is based on the Diffie–Hellman key exchange for public-key cryptography and is an asymmetric key algorithm. This scheme was introduced in 1985 by Taher El-Gamal. It is based on discrete logarithm problems and incorporates encryption and digital signature algorithms. This approach enhances the security of the cryptosystem. The EI-GAMAL cryptosystem consists of three stages: key generation, encryption, and decryption.

4.2.1. Key generation phase

This stage initializes from the creation of a high prime order Q of the cyclic set $H = \{1, 2, ...Q - 1\}$ form that the generator h is elected. The private key of the receiver is b also chosen from H. From these public attributes, the receiver's public key c is estimated utilizing Equation (9).

$$c = h^b \bmod Q \tag{9}$$

The private key of the receiver is denoted as b, and this is secured privately in the receiver's possession. The integration of the public attributes is referred to as the public key. The public key of the sender is termed as $\{h, Q, c\}$.

4.2.2. Message encryption phase

In this stage, the sender detects the arbitrary integer y from the cyclic set H. Based on the public key $\{h, Q, c\}$ s, the message n is encrypted by the sender by evaluating $\{d_1, d_2\}$. Equation (10) for the public key and Equation (11) formulate for the ciphertext.

$$d_1 = h^y \bmod Q \tag{10}$$

$$d_2 = n * c^y \bmod Q \tag{11}$$

The integrated message is denoted as $\{d_1, d_2\}$, and it is forwarded to the receiver.

4.2.3. Decryption phase

In this stage, by extracting n utilizing Equation (12), the recipient decrypts the message $\{d_1, d_2\}$.

$$n = \frac{d_2}{d_1^b} \bmod Q \tag{12}$$

Here, the private key of the receiver is denoted as b, and the public key is indicated as Q. The plaintext message is termed as n.

Finally, the secured data is attained with the aid of the EI-GAMAL approach.

4.3. Adaptive EI-GAMAL for data encryption

The EPSRGO algorithm adjusts attributes and generates the key for data encryption and decryption. It is an enhanced version of the traditional GOA. EPSRGO optimizes the key for GI-GAMAL to reduce computation time and memory usage. The traditional GOA provides highly competitive solutions, making it the preferred choice for implementing the new algorithm in the decentralized big data auditing mechanism for blockchain-based cloud storage.

However, the conventional GOA requires improvements in convergence and execution time. Therefore, EPSRGO has been implemented in this mechanism. An adaptive strategy is introduced to optimize the key in the proposed encryption mechanism. Equation (13) defines the objective function.

$$ob = \arg \min_{\{ky^n\}} [tm + ms] \tag{13}$$

Here, the term ky^n refers to the key in the binary part with values 0 or 1. Also, the time and memory size are denoted as specified as tm *and* ms correspondingly.

Time: The overall period to complete the encryption task is referred to as time tm. It is formulated in Equation (14).

$$tm = \frac{di}{sp} \tag{14}$$

Here, the distance and speed are denoted as di *and* sp accordingly.

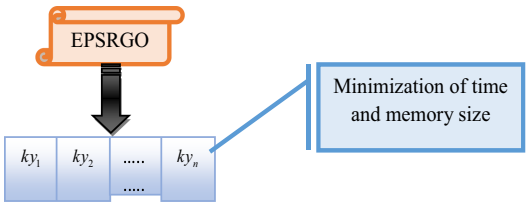
Memory size: The space that is needed to process the encryption is referred to as memory size ms. It is derived in Equation (15).

$$ms = lo \times si \tag{15}$$

Here, the data location and the word size are specified as lo *and* si correspondingly.

Figure 4 depicts the solution encoding diagram of the AEI-GAMAL approach.

Figure 4
The solution encoding diagram of the suggested adaptive EI-GAMAL task



5. An Intelligent Auditing Scheme of Big Data in Cloud Using Divide and Conquer Table

5.1. Location array and information

The D&CT approach is primarily used to find the optimal solution to the given problem. Typically, it decomposes the given problem into two or more subproblems and combines their solutions to resolve it. In D&CT, array data structures are used to store and process input data. Another key component is the information table, which contains the user ID, file ID, file details, and version number for each data block. These details are unique to each data owner and remain consistent during data modifications. Whenever data modifications occur, the location of the array and the information table vary. When data is inserted, the array and information table are updated. In contrast, their size decreases when the data owner deletes data.

5.2 Data storage using D&CT process

Data storage is a key feature of the data auditing mechanism. Fundamental operations such as insert, delete, modify, and append are used in cloud computing to update user data. Cloud computing requires the creation of an advanced data structure called D&CT to prevent data attacks and support data updates. The data owner constructs D&CT data structures before storing data blocks. D&CT has two primary components: the logical index and the version number.

5.2.1. Insert

In the data insertion process, new data is appended to the end of the data block. Initially, the system identifies the position of the last updated block and generates a new row adjacent to the final data entry. The data owner assigns the logical index and version number to the data block. The upper and lower boundaries of the D&CT are incremented by 1. Finally, the block tag is generated, and the data insertion is completed.

5.2.2. Delete

When the data owner needs to delete specific data, it is removed from the D&CTs. The upper and lower boundaries are decremented by 1 after deletion.

5.2.3. Update

During an update operation, the data owner is responsible for storing and managing the D&CT. The data structure facilitates the data auditing design in the proposed approach, ensuring consistent performance. However, it must maintain a similar structure during insertion and deletion operations. However, these approaches impose a high computational burden on the data owner. To enhance the data auditing process, data modification is essential. In this case, the data owner can modify the corresponding data block. Figure 5 illustrates the D&CT operations for cloud-based data auditing.

6. Results and Discussions

6.1. Experimental setup

The proposed decentralized big data auditing mechanism for blockchain-based cloud storage was implemented in Python, yielding promising results. The chromosome length was 16, and the population size was 10. In addition, the maximum number of iterations was set to 50. Various optimization approaches were employed to evaluate the proposed mechanism, including Harris Hawks Optimization (HHO)-AEI-GAMAL [46], Hydrological Cycle Algorithm (HCA)-AEI-GAMAL [47], Remora Optimization Algorithm (ROA)-AEI-GAMAL [48], Giant Trevally Optimizer (GTO)-AEI-GAMAL, Genetic Algorithm (GA)-AEI-GAMAL [49], Particle Swarm Optimization (PSO)-AEI-GAMAL [50], Black-Box Optimization (BBO)-AEI-GAMAL [51], Flower Pollination Algorithm (FPA)-AEI-GAMAL, Grey Wolf Optimization (GWO)-AEI-GAMAL, Bat Algorithm (BA)-AEI-GAMAL, Firefly Algorithm (FA)-AEI-GAMAL, Cuckoo Search Algorithm (CS)-AEI-GAMAL, Moth-Flame Optimization (MFO)-AEI-GAMAL, Gravitational Search Algorithm (GSA)-AEI-GAMAL, and Differential Evolution (DE)-AEI-GAMAL. Moreover, several conventional encryption algorithms, including Data Encryption Standard (DES), Advanced Encryption Standard (AES), Rivest-Shamir-Adleman (RSA) [52], and EI-GAMAL, were employed to validate the system's effectiveness.

6.2. Evaluation metrics

The factors contributing to the evaluation of the proposed decentralized big data auditing mechanism for blockchain-based cloud storage are outlined below.

6.2.1. Communication costs

Net costs refer to the expenses payable by the service provider to the network provider for delivering telecommunication services.

6.2.2. Computational cost

It measures the amount of resources the network utilizes for interference or training.

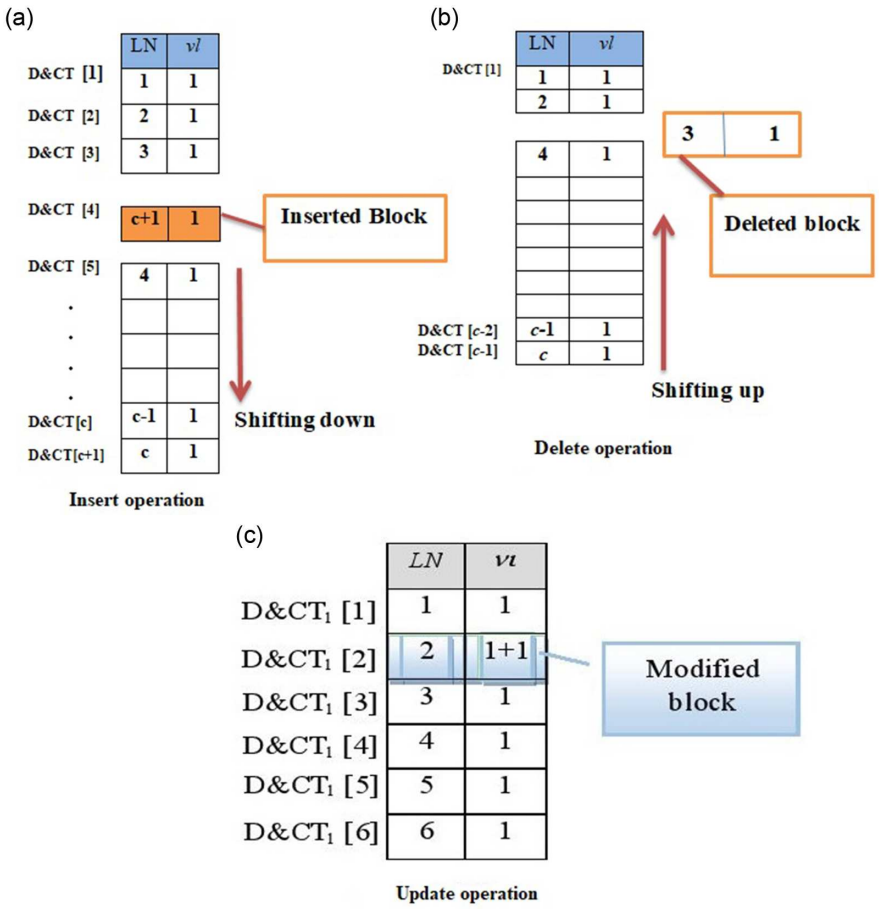
6.2.3. Chosen-plaintext attack (CPA)

It is a cryptanalysis attack that assumes the attacker can obtain the ciphertexts for arbitrary plaintexts.

6.2.4. Known-plaintext attack (KPA)

A cryptanalysis attack in which the attacker has access to both the plaintext and the corresponding ciphertext.

Figure 5
The operations of D&CT for the data auditing mechanism in the cloud



6.2.5. Encryption time

Used to measure the throughput of an encryption process.

6.2.6. Decryption time

Used to measure the time required to convert encrypted data back into its original form.

Reason for validating the developed model with traditional mechanisms in terms of diverse performance factors: The analysis of the developed model ensures that this mechanism is suitable for attaining efficiency, security, and scalability. Also, this mechanism does not contain significant drawbacks when compared to conventional techniques. The analysis of communication cost helps to ensure effective communication between users by solving unnecessary overloads with less network utilization. Also, when the computational cost of the developed model is tested with other techniques, the computational power utilized by the model can be verified, which also leads to ensuring the scalability and efficiency of the overall system. Through the CPA and KPA analysis, the security performance is tested to know the robustness of the strategy over these attacks and also its viability in practical environments. Finally, with the support of encryption and decryption time analysis, it can be known that the latency issues of the model can be identified, which helps to make use of the model in time-sensitive applications if it does not contain any latency issues.

6.3. Communication cost examination of the suggested decentralized big data auditing mechanism for cloud storage based on blockchain over diverse conventional algorithms and cryptography techniques

The communication cost against conventional cryptography approaches and algorithms is presented in Figures 6 and 7, respectively. By varying the blocks, the communication cost is evaluated for the new system. From Figure 6, the communication cost of the developed decentralized big data auditing mechanism for cloud

Figure 6
Communication cost examination of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent cryptography approaches

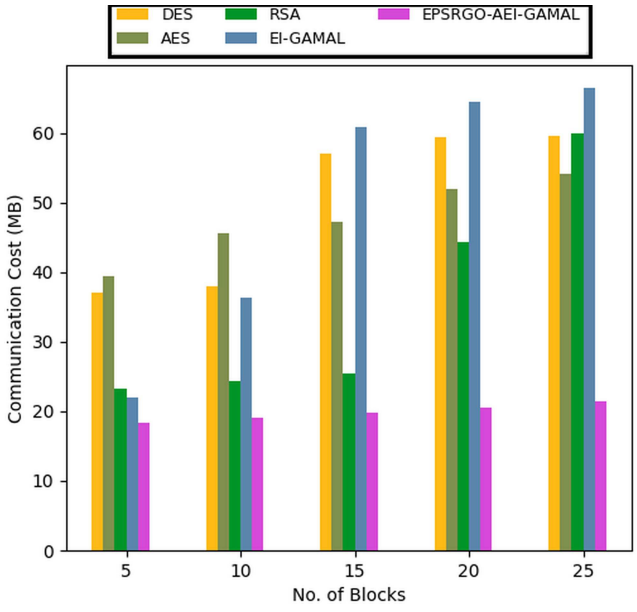
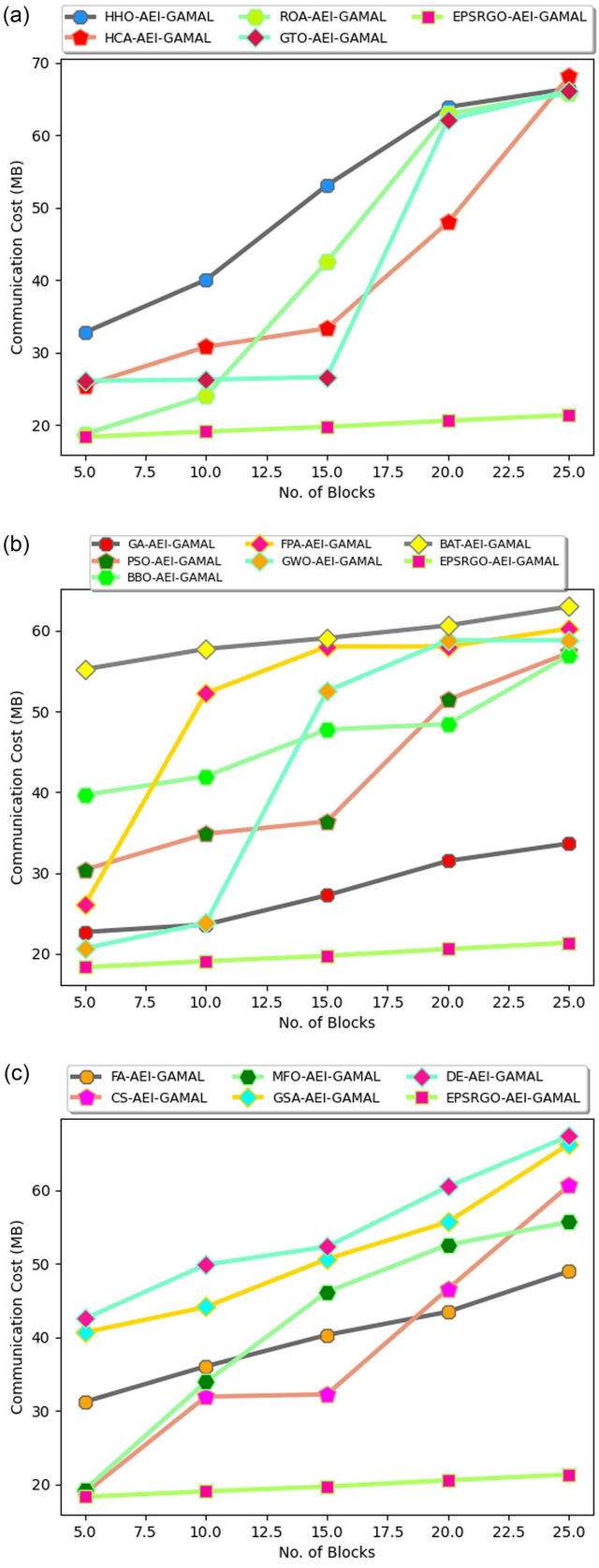


Figure 7
Communication cost of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent optimization algorithms



storage based on blockchain is minimized by 18.3% of DES, 15.6% of AES, 13% of RSA, and 20.6% of EI-GAMAL accordingly when the number of blocks is 10. This is attained because the auditing process is shared between various participants (auditors) instead of depending on a central authority. Here, the developed EPSRGO-AEI-GAMAL helps to transfer the significant information including the encrypted hashes for auditing and acquires less and optimal usage of network resources. Hence, it has been shown that the designed task has better functionality rates than the other classical approaches.

6.4. Computational cost validation of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over diverse traditional algorithms and cryptography approaches

The validation of the computational cost of the implemented decentralized big data auditing mechanism for cloud storage based on blockchain over multiple traditional cryptography approaches and algorithms is depicted in Figures 8 and 9 appropriately. The block size helped to validate the computational cost. When the number of blocks is 12.5, the computational cost of the suggested decentralized big data auditing mechanism for cloud storage based on blockchain is decreased by 14% of HHO-AEI-GAMAL, 8.4% of HCA-AEI-GAMAL, 15.6% of ROA-AEI-GAMAL, and 7.2% of GTO-AEI-GAMAL in Figure 9(a) correspondingly. These results have been attained with the support of the developed EPSRGO-AEI-GAMAL as it has the efficiency to handle well-established theoretical problems and reduces the operational complexity at the time of verification. This leads to the reduction of loads on the auditors and clients, resulting in lower computational cost for the system. Thus, it is confirmed that the suggested scheme has better performance rates than the pre-existing tasks.

Figure 8
Computational cost validation of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent cryptography approaches

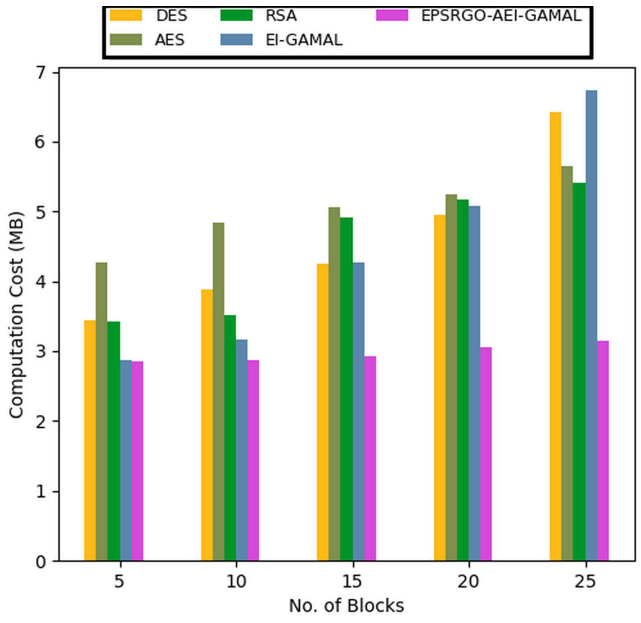
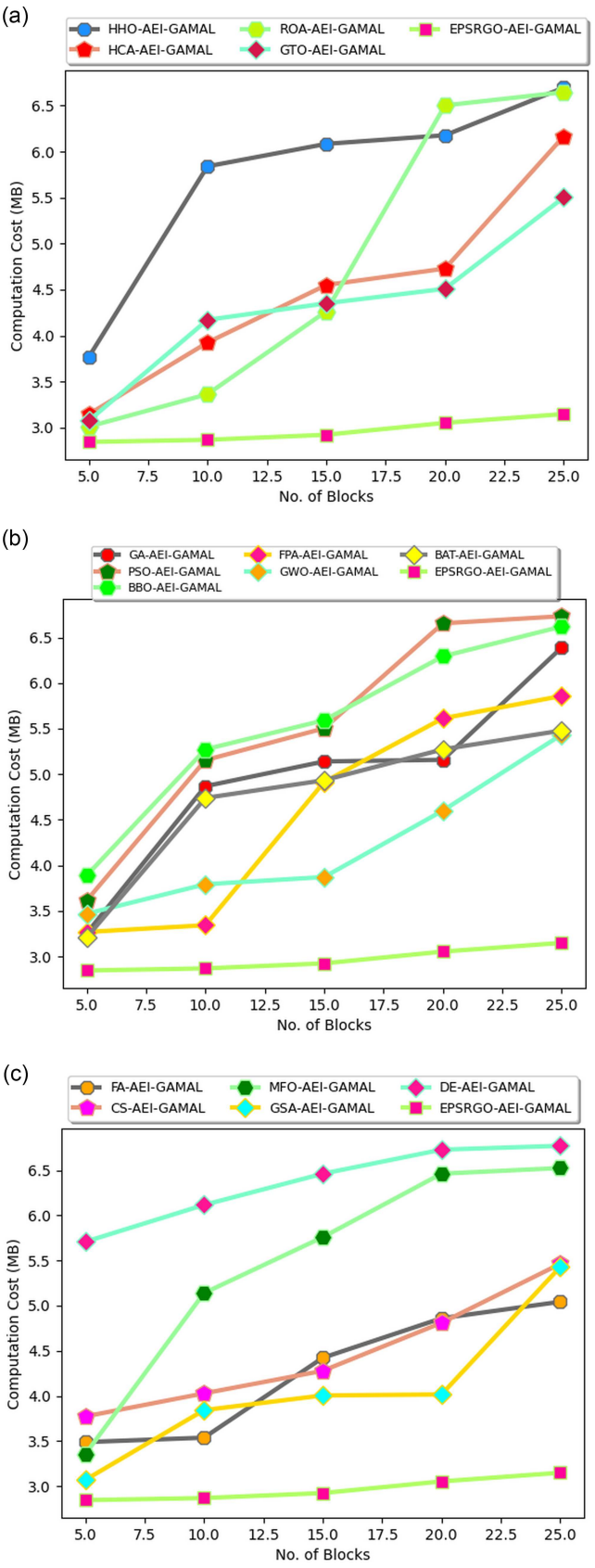


Figure 9
Computational cost validation of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent optimization algorithms



6.5. Convergence estimation of the developed EPSRGO algorithm over diverse conventional algorithms by varying the block sizes

Figure 10 illustrates the convergence validation of the improved EPSRGO algorithm with various optimization algorithms. By utilizing the iteration counts, the convergence is estimated for the suggested EPSRGO algorithm. When the iteration count is 40 for the suggested EPSRGO algorithm in Figure 10(a), the convergence is enriched by 97.42% of GA-AEI-GAMAL, 97.4% of PSO-AEI-GAMAL, 97.5% of BBO-AEI-GAMAL, 97.5% of FPA-AEI-GAMAL, 97.45% of GWO-AEI-GAMAL, and 97.38% of BA-AEI-GAMAL accordingly. From this experiment, it is revealed that the suggested EPSRGO algorithm has better convergence rates than the older mechanisms.

6.6. Statistical evaluation of the developed EPSRGO algorithm over diverse conventional algorithms by varying the block sizes

Table 2 elucidates the statistical investigation of the designed EPSRGO algorithm with traditional algorithms. The designed EPSRGO algorithm has improved by 13% of HHO-AEI-GAMAL, 12.8% of HCA-AEI-GAMAL, 25.4% of ROA-AEI-GAMAL, 5% of GTO-AEI-GAMAL, 4.2% of GA-AEI-GAMAL, 2.8% of PSO-AEI-GAMAL, 3.7% of BBO-AEI-GAMAL, 2.4% of FPA-AEI-GAMAL, 3.8% of GWO-AEI-GAMAL, 6.9% of BA-AEI-GAMAL, 9.2% of FA-AEI-GAMAL, 1.9% of CS-AEI-GAMAL, 5.3% of MFO-AEI-GAMAL, 1.9% of GSA-AEI-GAMAL, and 2.9% of DE-AEI-GAMAL correspondingly when taking the best measure. Therefore, it is demonstrated that the implemented EPSRGO algorithm has higher functionalities (Table 2).

6.7. Analysis of CPA for the developed decentralized big data auditing mechanism for cloud storage based on blockchain over diverse conventional algorithms and cryptography approaches

The CPA examination of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain was performed over various cryptography approaches and algorithms and depicted in Figures 11 and 12 accordingly. When focusing on the number of blocks as 7.5 in Figure 12(c), the CPA of the suggested decentralized big data auditing mechanism for cloud storage based on blockchain is reduced by 73.6% of FA-AEI-GAMAL, 73.7% of CS-AEI-GAMAL, 73.5% of MFO-AEI-GAMAL, 74.7% of GSA-AEI-GAMAL, and 73.6% of DE-AEI-GAMAL appropriately. The developed EPSRGO-AEI-GAMAL mechanism is resilient to CPA, as it maintains encryption efficiency even when attackers obtain partial plaintext–ciphertext pairs, preventing them from gathering

Figure 10
Convergence estimation of the suggested EPSRGO algorithm over divergent optimization algorithms

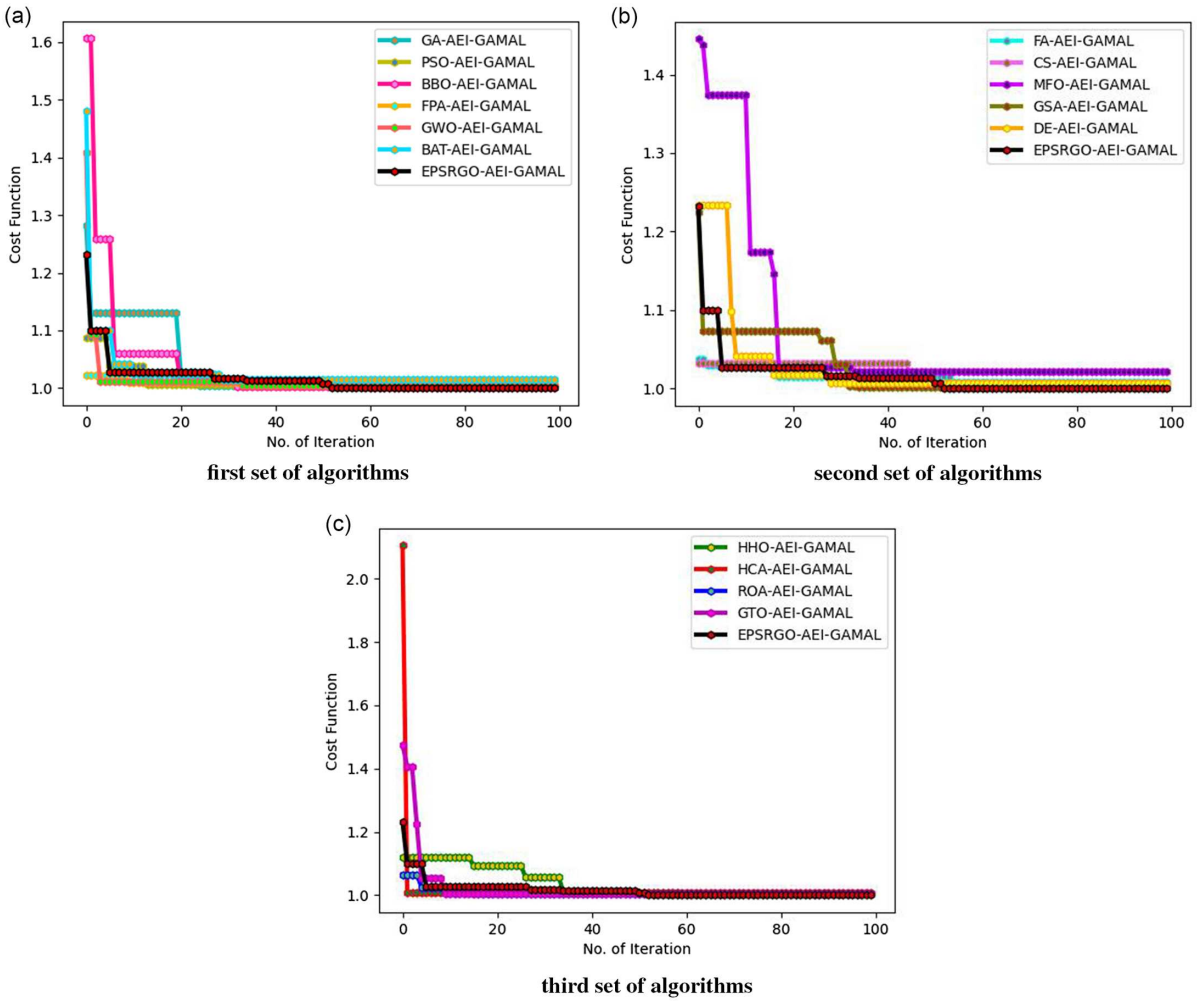
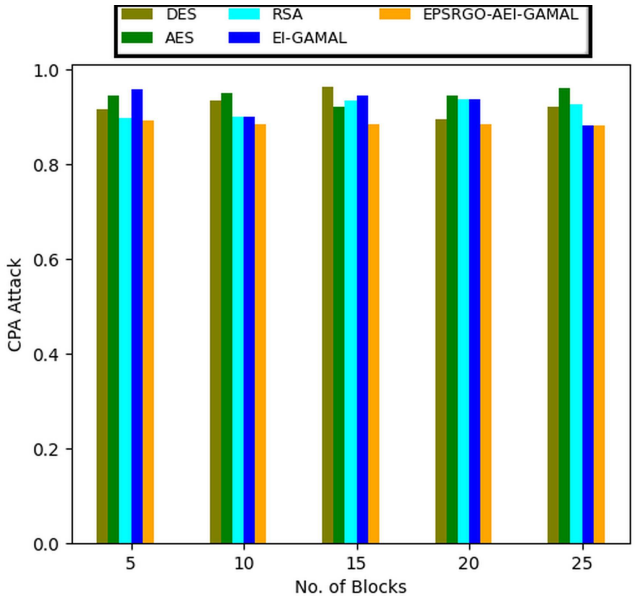


Table 2
Statistical investigation of the designed EPSRGO algorithm over divergent optimization algorithms

ALGORITHMS	Best	Worst	Mean	Median	Standard deviation
HHO-AEI-GAMAL	1.004193299	1.120411202	1.036826731	1.008905082	0.045412782
HCA-AEI-GAMAL	1.008924043	2.104381464	1.020016071	1.008924043	0.108983015
ROA-AEI-GAMAL	1.005378592	1.065167263	1.010038762	1.005378592	0.012963717
GTO-AEI-GAMAL	1.005252002	1.474594631	1.022665306	1.005252002	0.075732963
GA-AEI-GAMAL	1.005096771	1.28241375	1.032678414	1.005096771	0.055149064
PSO-AEI-GAMAL	1.003763233	1.087599839	1.012134418	1.006938588	0.019505159
BBO-AEI-GAMAL	1.000500602	1.606052337	1.033138819	1.002045065	0.097105588
FPA-AEI-GAMAL	1.01011602	1.022234561	1.011206688	1.01011602	0.003468105
GWO-AEI-GAMAL	1.004476603	1.40885592	1.012229954	1.004476603	0.041727713
BA-AEI-GAMAL	1.015682629	1.480283286	1.027108718	1.015682629	0.049260402
FA-AEI-GAMAL	1.008767248	1.037316269	1.014869733	1.015397758	0.007665665
CS-AEI-GAMAL	1.006560971	1.032669151	1.018307581	1.006560971	0.01298637
MFO-AEI-GAMAL	1.022415552	1.445176964	1.072019926	1.022415552	0.116202799
GSA-AEI-GAMAL	1.001934653	1.224768281	1.024543335	1.001934653	0.037026649
DE-AEI-GAMAL	1.006690381	1.234108441	1.027609801	1.006690381	0.058103653
EPSRGO-AEI-GAMAL	1.000040039	1.231814906	1.015816445	1.010465161	0.02995017

Figure 11
CPA analysis of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent cryptography approaches



sufficient information to compromise the encrypted data. This ensures that the suggested task has better efficacy.

6.8. Examination of KPA for the developed decentralized big data auditing mechanism for cloud storage based on blockchain over diverse conventional algorithms and cryptography approaches

Figures 13 and 14 show the KPA examination of the implemented decentralized big data auditing mechanism for cloud storage based on blockchain over diverse classical cryptography approaches and algorithms accordingly. When the number of blocks is 5 in Figure 13, the KPA of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain is minimized by 83% of DES, 82.6% of AES, 81.4% of RSA, and 82.2% of El-GAMAL, respectively. This can be justified theoretically as follows. The developed EPSRGO-AEI-GAMAL ensures the integrity and confidentiality of the auditing data by making effective encryption against KPA. Even if some plaintexts and ciphertexts are known, it is difficult for an attacker to derive information about the encryption keys or other plaintexts. From this, it is ensured that the suggested task has high effectiveness.

6.9. The encryption time validation of the developed decentralized big data auditing mechanism for cloud storage based on blockchain over diverse conventional algorithms and cryptography approaches

The research on the encryption time of the developed decentralized big data auditing mechanism for cloud storage based on the blockchain is conducted against conventional cryptography mechanisms and algorithms and is illustrated in Figures 15 and 16 appropriately. When taking the number of block sizes as 10 in Figure 16(b), the encryption time of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain decreased by 9% of GA-AEI-GAMAL, 10.7% of PSO-AEI-GAMAL, 11.7% of BBO-AEI-GAMAL, 11.8% of FPA-AEI-GAMAL, 8.2% of GWO-AEI-GAMAL, and 9.7% of BA-AEI-GAMAL accordingly. The implemented EPSRGO-AEI-GAMAL provides less encryption time due to efficient modular exponentiation, especially in decentralized environments. This shows the developed system's better functionalities.

6.10. The decryption time validation of the developed decentralized big data auditing mechanism for cloud storage based on blockchain over diverse conventional algorithms and cryptography approaches

From Figures 17 and 18, the investigation of decryption time in a designed decentralized big data auditing mechanism for cloud

Figure 12
CPA analysis of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent optimization algorithms

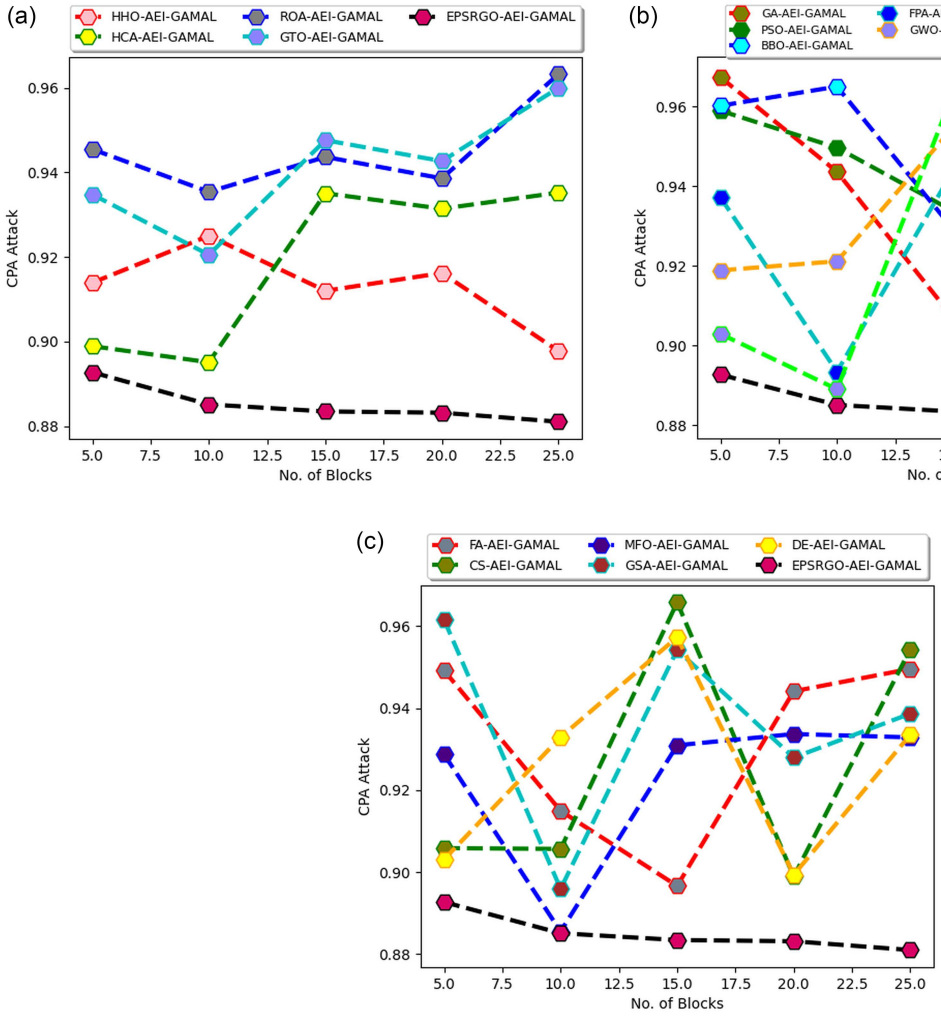


Figure 13
KPA analysis of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent cryptography approaches

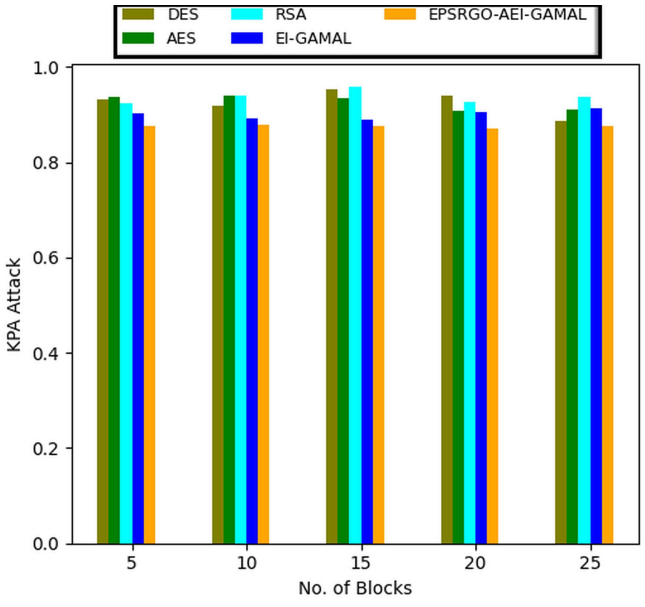


Figure 14
KPA analysis of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent optimization algorithms

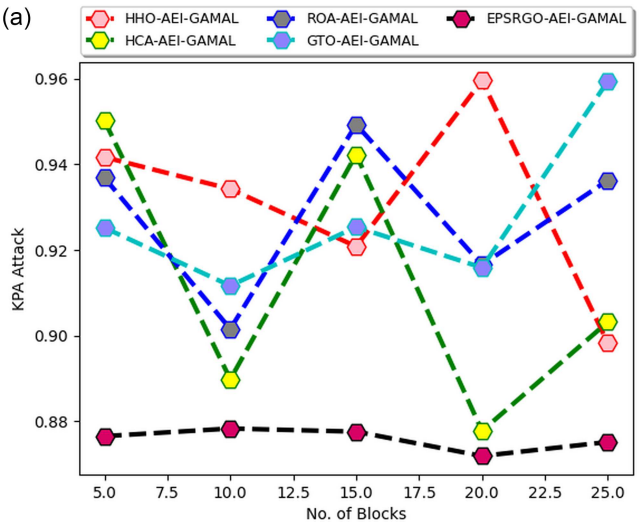


Figure 14
Continue

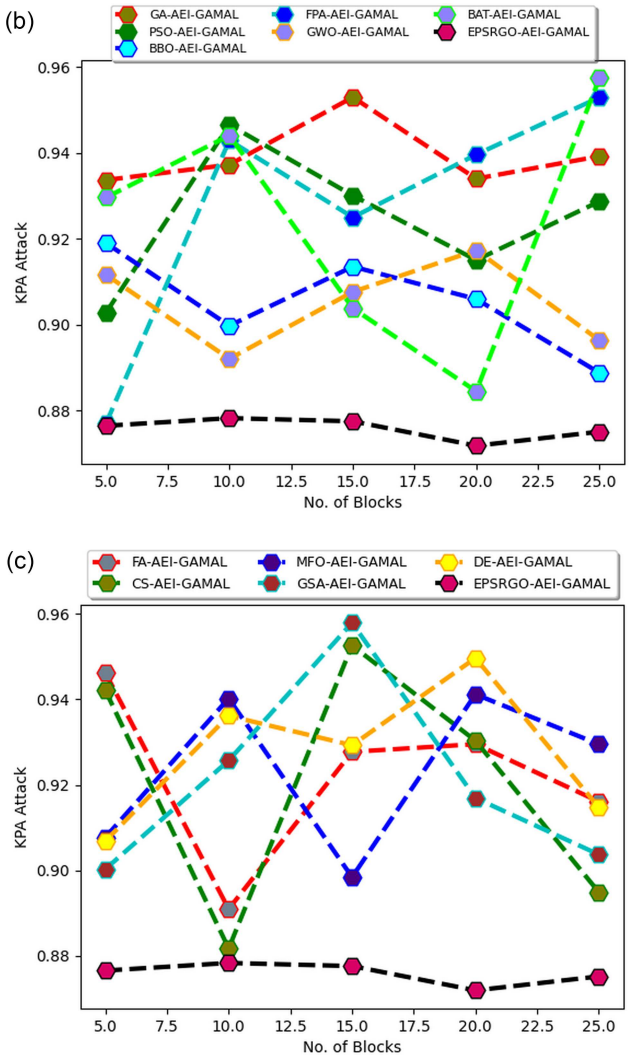


Figure 15
The encryption time validation of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent cryptography approaches

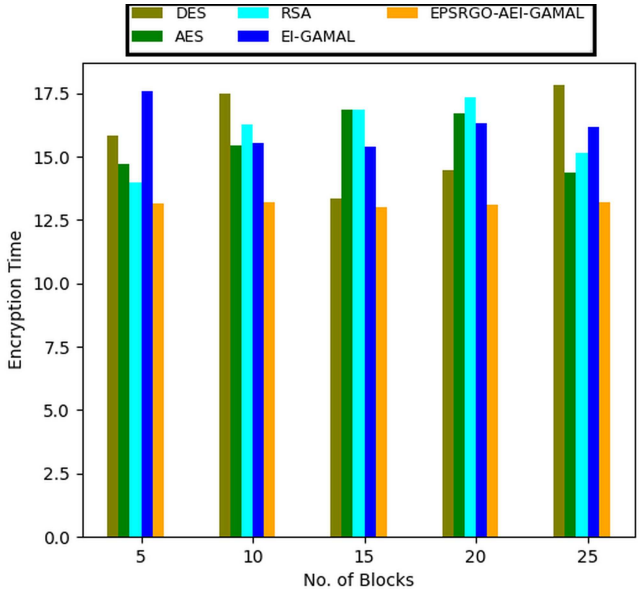


Figure 16
Encryption time validation of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent optimization algorithms

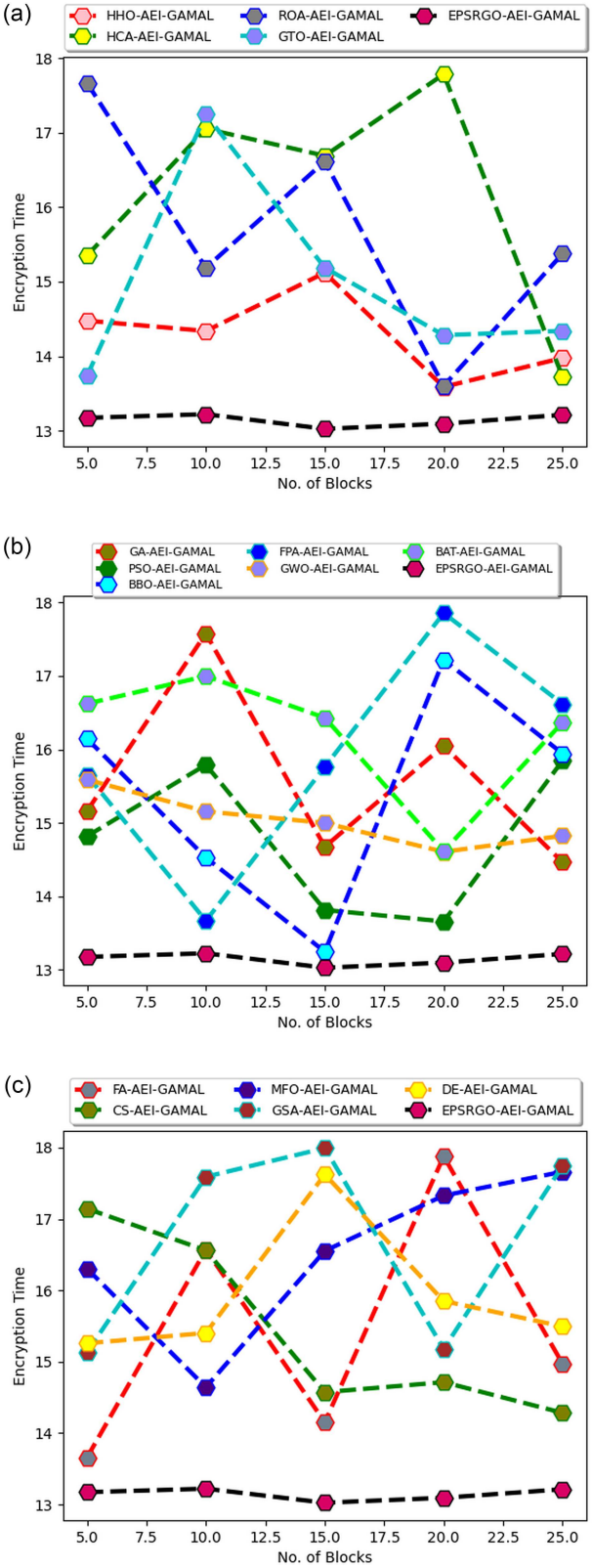
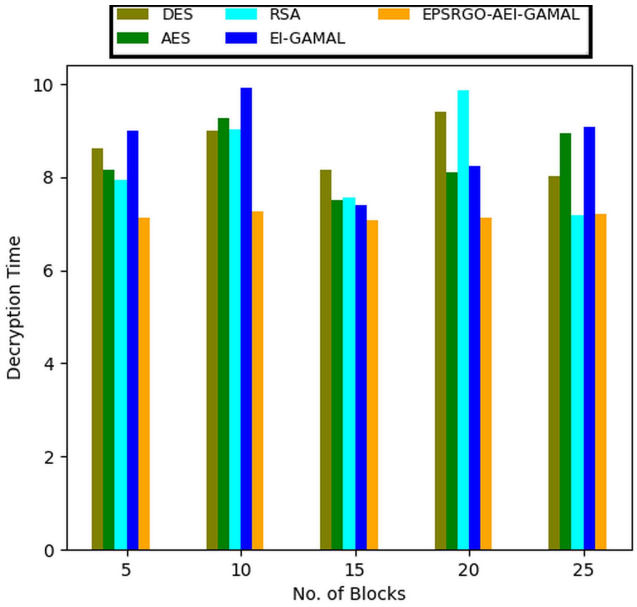


Figure 17
The decryption time validation of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent cryptography approaches



storage based on blockchain is performed over divergent cryptography approaches and algorithms correspondingly. From Figure 18(b), the decryption time of the new decentralized big data auditing mechanism for cloud storage based on blockchain is minimized by 77.5% of GA-AEI-GAMAL, 69% of PSO-AEI-GAMAL, 63% of BBO-AEI-GAMAL, 57.5% of FPA-AEI-GAMAL, 66.5% of GWO-AEI-GAMAL, and 75.5% of BA-AEI-GAMAL appropriately when the block size is 15. The implemented EPSRGO-AEI-GAMAL provides efficient decryption for auditing purposes, where only small portions of the data are decrypted. This explains why the implemented scheme has higher performance rates than the pre-existing approaches in terms of decryption time.

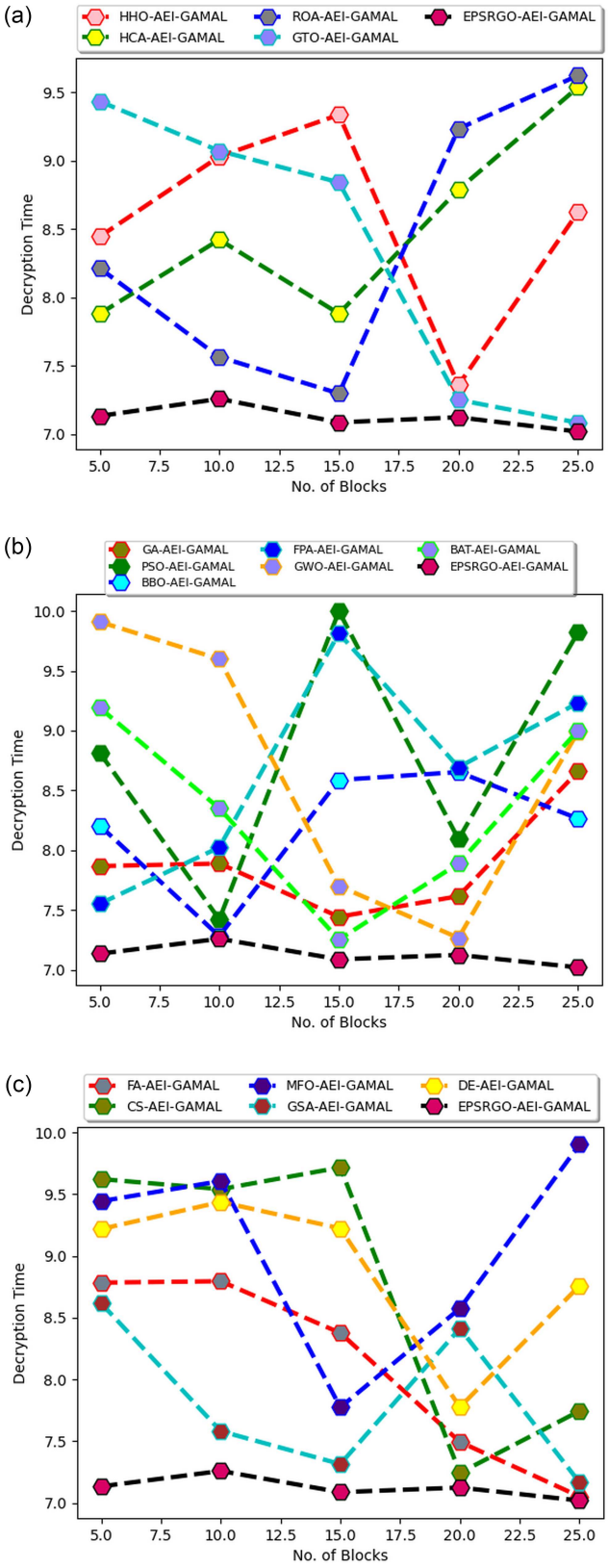
7. Conclusion

The advanced data auditing mechanism for cloud storage has been developed to enhance the reliability and scalability of data auditing solutions. It does not require a TPA because cloud providers collaboratively validate data, making it a decentralized blockchain framework.

First, essential data was gathered from benchmark datasets and divided into blocks for encryption. This encryption process was performed using the AEI-GAMAL algorithm with the support of the proposed EPSRGO. Subsequently, the encrypted data was stored in the cloud using the D&CT concept, where the data table and data array were updated to integrate with the data auditing mechanism. The data array included the file ID, version number, file data, and user ID, which remained consistent for every block. The version number varied based on deletions and updates.

Each file region contained a location table that recorded the region of every file and was updated whenever the D&CT was modified. This approach ensured the integrity of sensitive data in the cloud. The functionality of the proposed decentralized public auditing strategy for data storage was validated against traditional

Figure 18
Decryption time validation of the recommended decentralized big data auditing mechanism for cloud storage based on blockchain over divergent optimization algorithms



mechanisms based on various performance factors. As shown in the experiment, the KPA vulnerability of the proposed decentralized big data auditing mechanism for blockchain-based cloud storage was reduced by 69% compared to FA-AEI-GAMAL, 68.13% compared to CS-AEI-GAMAL, 70% compared to MFO-AEI-GAMAL, 68% compared to GSA-AEI-GAMAL, and 69% compared to DE-AEI-GAMAL when considering the number of blocks. These results confirm that the implemented data auditing strategy for blockchain-based cloud storage achieves higher effectiveness than existing techniques.

While the proposed decentralized data auditing mechanism for blockchain-based cloud storage has demonstrated significant improvements in security and performance, several areas warrant further investigation:

1) Scalability in Large-Scale Systems: Although the current study presents promising results in terms of security and integrity, further research is needed to evaluate the scalability of this blockchain-based auditing system in larger, more complex cloud environments. Examining its performance when handling massive datasets or numerous concurrent users will help identify potential bottlenecks and areas for optimization.
2) Integration with Emerging Cloud Architectures: Future research could explore integrating this auditing mechanism with modern cloud architectures, such as hybrid clouds or edge computing environments. Investigating its adaptability to distributed computing models could provide deeper insights into its broader applicability.

Addressing these areas will help further optimize blockchain-based data auditing mechanisms, enhancing their efficiency and adaptability to the evolving demands of cloud storage environments.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data are available on request from the corresponding author upon reasonable request.

Author Contribution Statement

Showri Rayalu Bandanadam: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Prasanna Kumar Rangarajan:** Supervision, Project administration.

References

[1] Shang, T., Zhang, F., Chen, X., Liu, J., & Lu, X. (2021). Identity-based dynamic data auditing for big data storage. *IEEE Transactions on Big Data*, *7*(6), 913–921. https://doi.org/10.1109/TBDATA.2019.2941882

[2] Fu, A., Yu, S., Zhang, Y., Wang, H., & Huang, C. (2022). NPP: A new privacy-aware public auditing scheme for cloud data sharing with group users. *IEEE Transactions on Big Data*, *8*(1), 14–24. https://doi.org/10.1109/TBDATA.2017.2701347

[3] Deng, L., Yang, B., & Wang, X. (2020). A lightweight identity-based remote data auditing scheme for cloud storage. *IEEE Access*, *8*, 206396–206405. https://doi.org/10.1109/ACCESS.2020.3037696

[4] Yang, X., Pei, X., Wang, M., Li, T., & Wang, C. (2020). Multi-replica and multi-cloud data public audit scheme based on blockchain. *IEEE Access*, *8*, 144809–144822. https://doi.org/10.1109/ACCESS.2020.3014510

[5] Bandanadam, S. R., & Kumar, R. P. (2022). A methodical literature survey on blockchain-based public auditing in cloud: Analysis on performance and door towards future scope. In *2022 International Conference on Applied Artificial Intelligence and Computing*, 1429–1436. https://doi.org/10.1109/ICAAIC53929.2022.9793311

[6] Zhou, L., Fu, A., Yang, G., Wang, H., & Zhang, Y. (2022). Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics. *IEEE Transactions on Dependable and Secure Computing*, *19*(2), 1118–1132. https://doi.org/10.1109/TDSC.2020.3013927

[7] Zhao, C., Xu, L., Li, J., Wang, F., & Fang, H. (2019). Fuzzy identity-based dynamic auditing of big data on cloud storage. *IEEE Access*, *7*, 160459–160471. https://doi.org/10.1109/ACCESS.2019.2950938

[8] Wang, F., Xu, L., Li, J., & Choo, K. K. R. (2022). Lightweight public/private auditing scheme for resource-constrained end devices in cloud storage. *IEEE Transactions on Cloud Computing*, *10*(4), 2704–2716. https://doi.org/10.1109/TCC.2020.3045806

[9] Li, H., Liu, L., Lan, C., Wang, C., & Guo, H. (2020). Lattice-based privacy-preserving and forward-secure cloud storage public auditing scheme. *IEEE Access*, *8*, 86797–86809. https://doi.org/10.1109/ACCESS.2020.2991579

[10] Kumar, R. P., & Bandanadam, S. R. (2024). Block chain-based decentralized public auditing for cloud storage with improved EIGAMAL encryption model. *International Journal of Information Technology*, *16*(2), 697–711. https://doi.org/10.1007/s41870-023-01599-8

[11] Sun, Y., Liu, Q., Chen, X., & Du, X. (2020). An adaptive authenticated data structure with privacy-preserving for big data stream in cloud. *IEEE Transactions on Information Forensics and Security*, *15*, 3295–3310. https://doi.org/10.1109/TIFS.2020.2986879

[12] Yang, X., Wang, M., Wang, X., Chen, G., & Wang, C. (2020). Stateless cloud auditing scheme for non-manager dynamic group data with privacy preservation. *IEEE Access*, *8*, 212888–212903. https://ieeexplore.ieee.org/document/9266794

[13] Zhang, Y., Yu, J., Hao, R., Wang, C., & Ren, K. (2020). Enabling efficient user revocation in identity-based cloud storage auditing for shared big data. *IEEE Transactions on Dependable and Secure Computing*, *17*(3), 608–619. https://doi.org/10.1109/TDSC.2018.2829880

[14] Kumar, P., & Shah, M. (2020). To build scalable and portable blockchain application using Docker. In *Soft Computing: Theories and Applications: Proceedings of SoCTA 2019*, 619–628. https://doi.org/10.1007/978-981-15-4032-5_56

[15] Sookhak, M., Yu, F. R., & Zomaya, A. Y. (2018). Auditing big data storage in cloud computing using divide and conquer tables. *IEEE Transactions on Parallel and Distributed Systems*, *29*(5), 999–1012. https://doi.org/10.1109/TPDS.2017.2784423

[16] Dai, W., Tuo, S., Yu, L., Choo, K. K. R., Zou, D., & Jin, H. (2022). HAPPS: A hidden attribute and privilege-protection data-sharing scheme with verifiability. *IEEE Internet of Things Journal*, *9*(24), 25538–25550. https://doi.org/10.1109/JIOT.2022.3197708

[17] Yang, X., Wang, M., Li, T., Liu, R., & Wang, C. (2020). Privacy-preserving cloud auditing for multiple users scheme with authorization and traceability. *IEEE Access*, *8*, 130866–130877. https://doi.org/10.1109/ACCESS.2020.3009539

[18] Zheng, W., Lai, C. F., He, D., Kumar, N., & Chen, B. (2021). Secure storage auditing with efficient key updates for cognitive industrial IoT environment. *IEEE Transactions on Industrial Informatics*, *17*(6), 4238–4247. https://doi.org/10.1109/TII.2020.2991204

[19] Yu, Y., Li, Y., Ni, J., Yang, G., Mu, Y., & Susilo, W. (2016). Comments on "public integrity auditing for dynamic data sharing with multiuser modification." *IEEE Transactions on Information Forensics and Security*, *11*(3), 658–659. https://doi.org/10.1109/TIFS.2015.2501728

[20] Chishti, M. S., Sufyan, F., & Banerjee, A. (2022). Decentralized on-chain data access via smart contracts in Ethereum blockchain. *IEEE Transactions on Network and Service Management*, *19*(1), 174–187. https://doi.org/10.1109/TNSM.2021.3120912

[21] Mohan, A. P., Asfak, M. R., & Gladston, A. (2020). Merkle tree and blockchain-based cloud data auditing. *International Journal of Cloud Applications and Computing*, *10*(3), 54–66. https://doi.org/10.4018/IJCAC.2020070103

[22] Yu, H., Yang, Z., & Sinnott, R. O. (2019). Decentralized big data auditing for smart city environments leveraging blockchain technology. *IEEE Access*, *7*, 6288–6296. https://doi.org/10.1109/ACCESS.2018.2888940

[23] Liu, B., Zhang, X., Yang, X., Zhang, Y., Xue, J., & Zhou, R. (2025). Blockchain-assisted fine-grained deduplication and integrity auditing for outsourced large-scale data in cloud storage. *IEEE Internet of Things Journal*. Advance online publication. https://doi.org/10.1109/JIOT.2025.3548681

[24] Liu, C., Ranjan, R., Yang, C., Zhang, X., Wang, L., & Chen, J. (2015). MuR-DPA: Top-down levelled multi-replica Merkle hash tree based secure public auditing for dynamic big data storage on cloud. *IEEE Transactions on Computers*, *64*(9), 2609–2622. https://doi.org/10.1109/TC.2014.2375190

[25] Lekshmi, M. M., & Subramanian, N. (2020). Data auditing in cloud storage using smart contract. In *Third International Conference on Smart Systems and Inventive Technology*, 999–1002. https://doi.org/10.1109/ICSSIT48917.2020.9214112

[26] Shu, J., Zou, X., Jia, X., Zhang, W., & Xie, R. (2022). Blockchain-based decentralized public auditing for cloud storage. *IEEE Transactions on Cloud Computing*, *10*(4), 2366–2380. https://doi.org/10.1109/TCC.2021.3051622

[27] Fan, K., Bao, Z., Liu, M., Vasilakos, A. V., & Shi, W. (2020). Dredas: Decentralized, reliable and efficient remote outsourced data auditing scheme with blockchain smart contract for industrial IoT. *Future Generation Computer Systems*, *110*, 665–674. https://doi.org/10.1016/j.future.2019.10.014

[28] Li, J., Wu, J., Jiang, G., & Srikanthan, T. (2020). Blockchain-based public auditing for big data in cloud storage. *Information Processing & Management*, *57*(6), 102382. https://doi.org/10.1016/j.ipm.2020.102382

[29] Zhang, Q., Qian, S., Cui, J., Zhong, H., Wang, F., & He, D. (2025). Blockchain-based privacy-preserving deduplication and integrity auditing in cloud storage. *IEEE Transactions on Computers*, *74*(1), 123–135. https://doi.org/10.1109/TC.2025.3540670

[30] Wang, H., Wang, X. A., Xiao, S., & Liu, J. (2021). Decentralized data outsourcing auditing protocol based on blockchain. *Journal of Ambient Intelligence and Humanized Computing*, *12*(2), 2703–2714. https://doi.org/10.1007/s12652-020-02432-x

[31] Mishra, R., Ramesh, D., Edla, D. R., & Trivedi, M. C. (2022). Blockchain-assisted privacy-preserving public auditable model for cloud environment with efficient user revocation. *Cluster Computing*, *25*(5), 3103–3127. https://doi.org/10.1007/s10586-021-03508-9

[32] Liu, Z., Ren, L., Feng, Y., Wang, S., & Wei, J. (2023). Data integrity audit scheme based on quad Merkle tree and blockchain. *IEEE Access*, *11*, 59263–59273. https://doi.org/10.1109/ACCESS.2023.3240066

[33] Yu, H., Lu, X., & Pan, Z. (2020). An authorized public auditing scheme for dynamic big data storage in cloud computing. *IEEE Access*, *8*, 151465–151473. https://doi.org/10.1109/ACCESS.2020.3016760

[34] Manikumar, D. V. V. S., & Maheswari, B. U. (2020). Blockchain-based DDoS mitigation using machine learning techniques. In *Second International Conference on Inventive Research in Computing Applications*, 794–800. https://doi.org/10.1109/ICIRCA48905.2020.9183092

[35] Mageshwari, M., & Naresh, R. (2023). Improved sunflower optimization algorithm based encryption with public auditing scheme in secure cloud computing. *International Journal of Intelligent Engineering & Systems*, *16*(6), 13–23. https://doi.org/10.22266/ijies2023.1231.02

[36] Rani, P. S., & Priya, S. B. (2023). Security-aware and privacy-preserving blockchain chameleon hash functions for education system. *ECTI Transactions on Computer and Information Technology*, *17*(2), 225–234. https://doi.org/10.37936/ecti-cit.2023172.252014

[37] Poorvaja, S., Poojasree, V., Pon Anitha, S., & Baghavathi Priya, S. (2019). Blockchain-based certificate validation. In *National Conference on Data Science and Intelligent Information Technology*.

[38] Rajasoundaran, S., Santhosh Kumar, S. V. N., Selvi, M., Ganapathy, S., Rakesh, R., & Kannan, A. (2021). Machine learning based volatile blockchain construction for secure routing in decentralized military sensor networks. *Wireless Networks*, *27*(7), 4513–4534. https://doi.org/10.1007/s11276-021-02748-2

[39] Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2023). Gazelle optimization algorithm: A novel nature-inspired meta-heuristic optimizer. *Neural Computing and Applications*, *35*(5), 4099–4131. https://doi.org/10.1007/s00521-022-07854-6

[40] Ordonez, A. J., Medina, R. P., & Gerardo, B. D. (2018). Modified El Gamal algorithm for multiple senders and single receiver encryption. In *IEEE Symposium on Computer Applications & Industrial Electronics*, 201–205. https://doi.org/10.1109/ISCAIE.2018.8405470

[41] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, *97*, 849–872. https://doi.org/10.1016/j.future.2019.02.028

[42] Wang, S., Hussien, A. G., Jia, H., Abualigah, L., & Zheng, R. (2022). Enhanced remora optimization algorithm for solving constrained engineering optimization problems. *Mathematics*, *10*(10), 1696. https://doi.org/10.3390/math10101696

[43] Sadeeq, H. T., & Abdulazeez, A. M. (2022). Giant trevally optimizer (GTO): A novel metaheuristic algorithm for global optimization and challenging engineering problems. *IEEE Access*, *10*, 121615–121640. https://doi.org/10.1109/ACCESS.2022.3223388

[44] Venter, G., & Sobieszczanski-Sobieski, J. (2003). Particle swarm optimization. *AIAA Journal*, *41*(8), 1583–1589. https://arc.aiaa.org/doi/10.2514/2.2111

[45] Terayama, K., Sumita, M., Tamura, R., & Tsuda, K. (2021). Black-box optimization for automated discovery. *Accounts of Chemical Research*, *54*(6), 1334–1346. https://pubs.acs.org/doi/10.1021/acs.accounts.0c00713

[46] Yang, X. S., Karamanoglu, M., & He, X. (2014). Flower pollination algorithm: A novel approach for multiobjective optimization. *Engineering Optimization*, *46*(9), 1222–1237. https://doi.org/10.1080/0305215X.2013.832237

[47] Yang, X. S., & Gandomi, A. H. (2012). Bat algorithm: A novel approach for global engineering optimization. *Engineering Computations*, *29*(5), 464–483. http://dx.doi.org/10.1108/02644401211235834

[48] Johari, N. F., Zain, A. M., Mustaffa, N. H., & Udin, A. (2013). Firefly algorithm for optimization problem. *Applied Mechanics and Materials*, *421*, 512–517. https://doi.org/10.4028/www.scientific.net/AMM.421.512

[49] Mareli, M., & Twala, B. (2018). An adaptive Cuckoo search algorithm for optimization. *Applied Computing and Informatics*, *14*(2), 107–115. https://doi.org/10.1016/j.aci.2017.09.001

[50] Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

[51] Price, K. V. (2013). Differential evolution. In I. Zelinka, V. Snášel, & A. Abraham (Eds.), *Handbook of optimization: From classical to modern approach* (pp. 187–214). Springer. https://doi.org/10.1007/978-3-642-30504-7_8

[52] Neela, K. L., & Kavitha, V. (2022). An improved RSA technique with efficient data integrity verification for outsourcing database in cloud. *Wireless Personal Communications*, *123*(3), 2431–2448. https://doi.org/10.1007/s11277-021-09248-8

How to Cite: Bandanadam, S. R., & Rangarajan, P. K. (2025). Decentralized Big Data Auditing Scheme for Cloud Storage Based on Blockchain with Adaptive EI-GAMAL and Gazelle Optimization. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE52025101