Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064247

ARTICLE

C-BIVM: A Cognitive-Based Integrity Verification Model for IoT-Driven Smart Cities

Radhika Kumari[1], Kiranbir Kaur[1], Ahmad Almogren[2], Ayman Altameem[3], Salil Bharany[4,*], Yazeed Yasin Ghadi[5] and Ateeq Ur Rehman[6,*]

[1]Department of Computer Engineering & Technology, Guru Nanak Dev University, Punjab, 143005, India
[2]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, 11633, Saudi Arabia
[3]Department of Natural and Engineering Sciences, College of Applied Studies and Community Services, King Saud University, Riyadh, 11543, Saudi Arabia
[4]Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, India
[5]Department of Computer Science and Software Engineering, Al Ain University, Al Ain, 12555, Abu Dhabi
[6]School of Computing, Gachon University, Seongnam-Si, 13120, Republic of Korea

*Corresponding Authors: Salil Bharany. Email: salil.bharany@gmail.com; Ateeq Ur Rehman. Email: 202411144@gachon.ac.kr

Received: 10 February 2025; Accepted: 17 June 2025; Published: 30 July 2025

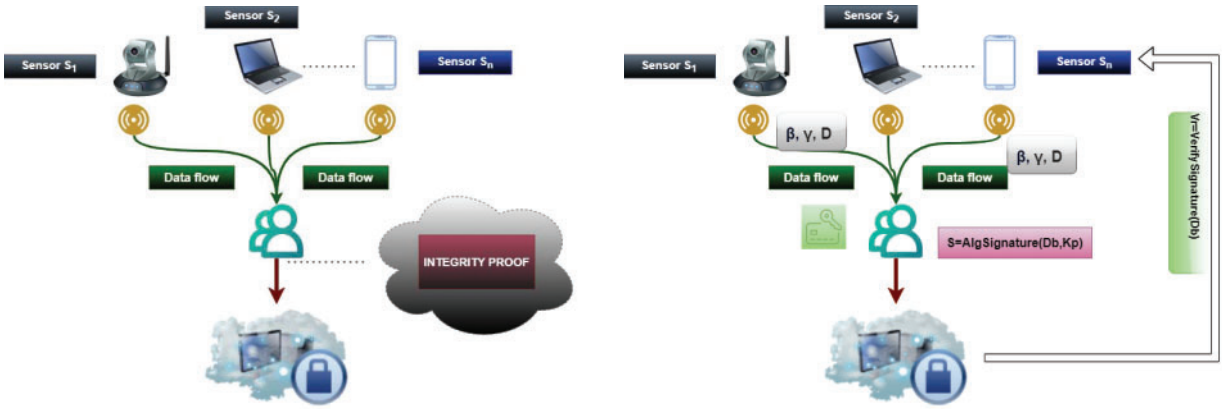
ABSTRACT: The exponential growth of the Internet of Things (IoT) has revolutionized various domains such as healthcare, smart cities, and agriculture, generating vast volumes of data that require secure processing and storage in cloud environments. However, reliance on cloud infrastructure raises critical security challenges, particularly regarding data integrity. While existing cryptographic methods provide robust integrity verification, they impose significant computational and energy overheads on resource-constrained IoT devices, limiting their applicability in large-scale, real-time scenarios. To address these challenges, we propose the Cognitive-Based Integrity Verification Model (C-BIVM), which leverages Belief-Desire-Intention (BDI) cognitive intelligence and algebraic signatures to enable lightweight, efficient, and scalable data integrity verification. The model incorporates batch auditing, reducing resource consumption in large-scale IoT environments by approximately 35%, while achieving an accuracy of over 99.2% in detecting data corruption. C-BIVM dynamically adapts integrity checks based on real-time conditions, optimizing resource utilization by minimizing redundant operations by more than 30%. Furthermore, blind verification techniques safeguard sensitive IoT data, ensuring privacy compliance by preventing unauthorized access during integrity checks. Extensive experimental evaluations demonstrate that C-BIVM reduces computation time for integrity checks by up to 40% compared to traditional bilinear pairing-based methods, making it particularly suitable for IoT-driven applications in smart cities, healthcare, and beyond. These results underscore the effectiveness of C-BIVM in delivering a secure, scalable, and resource-efficient solution tailored to the evolving needs of IoT ecosystems.

KEYWORDS: Internet of Things (IoT); smart cities; data integrity verification; BDI cognitive intelligence; algebraic signatures; batch auditing; resource-constrained devices; blind verification

1 Introduction

As the Internet of Things (IoT) increasingly drives smart city applications, the massive data generated by interconnected devices requires secure and trustworthy storage solutions. Cloud services often provide scalable storage for this data, but ensuring data integrity and privacy remains a critical challenge. Traditional data auditing methods, which rely on third-party auditors, may be vulnerable to malicious insiders or

attackers who can manipulate audit results or compromise data verification processes [1]. Decision-making in domains like transportation and energy management is based on this IoT data. However, since corruption or malicious alteration can potentially risk the city's services, infrastructure and public safety, it is necessary to ensure the integrity of the data [2,3]. Although IoT devices have locally limited computational and storage power, they rely heavily on cloud platforms, which may scale but have security challenges around data integrity [4]. Cloud-based smart city systems contain sensitive data, and their breach can cause severe disruption of essential services like traffic management and health care services [5,6]. Fig. 1a presents the essential components for preserving data integrity in a typical IoT-cloud environment. Data outsourcing makes it crucial to implement an integrity verification mechanism to ensure trust [7]. Additionally, data integrity becomes even more challenging to manage with the evolving landscape of IoT networks, where data is routinely sent and stored. Core cryptographic systems, such as bilinear pairing, yield beneficial, well-proven outcomes; however, they are expensive for computation and not acceptable for IoT [8]. This challenge leads to the need for low-cost and flexible integrity verification schemes in the context of smart city environments [9]. Likewise, the fact that routing protocols do not have strong validation mechanisms, which could allow them to malfunction and corrupt data systems, makes it vital to validate the integrity of the data during transfer. The existing gap in today's systems can be filled by the proposed Cognitive-Based Integrity Verification Model (C-BIVM) that integrates Belief-Desire-Intention (BDI) cognitive intelligence in the context of IoT networks. With C-BIVM, IoT devices can dynamically decide if, when, and how to perform integrity checks by monitoring the real-time environment and measuring the importance of data, significantly reducing unnecessary computations, while being resource-wise effective to the maximum extent [10,11]. Fig. 1b depicts the C-BIVM architecture in which BDI cognitive layers are embedded into the IoT devices to enable autonomous integrity verification management. This architecture reduces the computational load as a check is only executed if a relevant factor indicates it, improving performance and energy usage, which is particularly important with IoT devices for smart cities [12]. In addition, C-BIVM employs algebraic signatures as a lightweight computational primitive-based construction of a strong atomic cryptography module to fulfil effective smart city objectives for participants in resource-limited environments. This model also includes a batch auditing approach to speed up large-scale data verification [13].



(**a**) Traditional method architecture (**b**) Proposed method architecture

Figure 1: Traditional vs. proposed architecture

The proposed C-BIVM enhances data integrity verification in IoT-based smart city environments through the following key contributions:

- Lightweight Integrity Verification: By utilizing algebraic signatures, C-BIVM reduces the computational burden of traditional cryptographic methods, making it well-suited for resource-constrained IoT devices that require real-time data processing.
- Scalable Batch Auditing: The proposed system supports batch auditing, allowing multiple integrity checks to be processed simultaneously. This ensures efficient verification of large-scale data streams, addressing the scalability challenges in smart city IoT networks.
- Privacy-Preserving Verification: C-BIVM integrates blind verification techniques to protect sensitive data, such as health records and energy consumption details, ensuring privacy and compliance with regulations like GDPR while preventing unauthorized access by cloud service providers.
- Adaptive and Context-Aware Integrity Checks: The model incorporates an improved data structure that dynamically adjusts integrity verification processes based on real-time network conditions. This adaptability ensures robust and efficient verification as IoT networks evolve.

Section 2 presents the related work and the gap in prior methods. Section 3 details the proposed method. Section 4 illustrates the results and details the theoretical analysis. Section 5 discusses the potential outcomes and Section 6 represents the conclusion and future work.

2 Related Work

With the rapid development of IoT, generating vast amounts of data, proper processing and storage solutions are crucial. The Algebraic Integrity Verification for Cloud-IoT (AIVCI) efficiently addresses public data integrity verification in cloud-IoT environments. It leverages homomorphic hash functions and algebraic signatures to ensure efficient auditing and secure data privacy. The AIVCI embraces batch auditing for Blockchain-as-a-Service (BaaS), allowing a bulk approach to verifying vast amounts of data blocks at once; blind tech ensures critical data is not read by auditors during BaaS audits. It further presents the improved divide and conquer table, enabling more efficient, secure, and scalable data management [10]. The Enhanced Digital Signature Algorithm (EDSA) is an elliptic curve public key operation, which improves the integrity of cloud database data. The EDSA therewith improves the process of both key generation, and relevant capabilities, giving computational speeds of calculation, which has been exceptionally useful on cloud frameworks, or as they handle multiple leading records [12]. Integrating IoT with cloud computing can enhance this by offloading data, improving processing and storage capacity. Ensuring data integrity and privacy is vital throughout the data life cycle, from source to destination. Methods to verify data legitimacy in the cloud must address these concerns. Cloud storage offers a scalable alternative to physical storage but faces security challenges from untrusted servers that threaten data integrity. Balancing data integrity with communication and computation costs is critical. This paper proposes an integrity verification model using elliptic curve digital signature algorithm (ECDSA) combined with symmetric encryption to ensure data privacy and resist malicious attacks such as forgery, replacement, and replay attacks [14].

The edge data integrity for inspection (EDI-S) method improves fault-tolerance in edge computing environments, ensuring data consistency across multiple servers. It aggregates digital signatures for simultaneous verification across edge servers, enhancing integrity checks and detecting data compromise in distributed systems [15]. A new data stream verification mechanism, Privacy-Preserving Adaptive Trapdoor Hash Authentication Tree (P-ATHAT), merges trapdoor hash functions and BLS signatures in a Merkle hash tree for real-time verification. P-ATHAT adapts to growing data structures, preserving privacy during third-party audits, with rigorous security and efficiency evaluations [16]. A new lightweight data integrity audit protocol for cloud computing is designed for applications requiring frequent updates, supporting dynamic

data operations and bilinear pairings. It ensures data integrity without exposing information to unauthorized groups, with a performance assessment indicating high efficiency, especially for low-processing clients [17].

Strong methods like AIVCI and EDSA have some limitations in WSNs. The system can be human-reviewed to ensure auditability, but does not adequately account for the constraints of low-powered IoT devices with limited computational resources, nor is it sufficiently dynamically adapt itself in certain edge-cases that can result inefficiencies. At the same time, EDSA has weaknesses in performing dynamic updates of data and lacks ability to manage resources wisely. Some approaches raise privacy considerations since they might reveal when auditing sensitive information to cloud providers.

Therefore, C-BIVM is proposed to resolve these issues as shown in Table 1. The proposed system adds a cognitive layer based upon the BDI framework, allowing IoT devices in smart cities to infer when they should consider an integrity check based on the information available as well as with respect to their computational capabilities. This intelligent decision-making mechanism ensures that only the needed computations are performed, thereby over-providing resources while also being responsive to the current situation. C-BIVM utilizes algebraic signatures to speed up verification and adopts a batch auditing scheme for massive data. It also includes auditable proof of burn techniques to protect sensitive data during the verification process. By addressing and overcoming many of the existing inherent challenges on formerly proposed solutions, C-BIVM offers a more intelligent, efficient and privacy-preserving solution to realise data integrity verification at the intersection of IoT and machine learning.

Table 1: Summary of methods, problems addressed, and limitations

Method	Problem addressed	Limitations
AIVCI [10]	Efficient data auditing with algebraic signatures	Lacks real-time adaptation; does not fully address IoT constraints
EDSA [12]	Enhanced digital signatures with elliptic curves	Inefficient for dynamic data; lacks intelligent resource management
EDI-S [15]	Integrity verification for edge data	Static; limited adaptability to changing data/network conditions
P-ATHAT [16]	Authentication for big data streams	Potential privacy issues during audits; limited dynamic handling
Cloud auditing protocol [17]	Efficient auditing with bilinear pairings	Static adaptation; may not suit clients with limited resources

3 Proposed Methodology

In this section, we propose our proposed C-BIVM in WSN using routing protocol, which introduces a novel and light framework of BDI based cognitive intelligence with an algebraic signature, enabling data integrity verification in IoT-cloud systems. In addition, The C-BIVM model can be implemented into the IoT routing protocol to preserve the correctness of data in the course of transmission. With cognitive intelligence of BDI embedded in the routing nodes, the system now has the ability to determine generally when and how to perform integrity checks based on the real-time status of the network. If a node detects that there is corruption in the data or a security threat, it can send cognitive feedback to reroute it. Moreover, the integrity of these messages can be verified using lightweight algebraic signatures, making our approach suitable for resource-constrained routing nodes.

The C-BIVM method is differentiated, in terms of embedding decision-making within IoT sensors through the incorporation of cognitive aspects based on BDI model. The model gives sensors autonomous, situational awareness and as conditions change in real-time, the model responds intelligently when verifying integrity only as needed. Also, using algebraic signatures, C-BIVM ensures lightweight integrity verification without having to retrieve the original data back from the cloud. We go over the full methodology including the algorithmic components, how it works as well as showing new empirical strength and distinctiveness from the model.

3.1 Algorithm 1: BDI-Based Data Integrity Verification

The BDI-Based Data Integrity Verification algorithm leverages BDI cognitive intelligence to autonomously determine when to initiate data integrity checks. This approach is designed to optimize resource usage by enabling sensors to make intelligent decisions based on their current operational state and environmental conditions.

Algorithm 1: BDI-based data integrity verification

Input: IoT Sensor Data D, Current State S, Beliefs B, Desires D_e, Intentions I
Output: Verification Status V_s
$B \leftarrow$ Sensor's Current Knowledge of Data State
$D_e \leftarrow$ Desired Optimization (e.g., resource usage, security)
$I \leftarrow$ Plan for Verifying Data Integrity
if B is consistent with D_e and I **then**
    $V_s \leftarrow$ **Verified**
    Update $S \leftarrow S + 1$
    return V_s
else
    Abort Verification
    return "No Verification Needed"
end if

The BDI cognitive intelligence framework is implemented within IoT sensors to enable adaptive and intelligent decision-making for data integrity verification. The BDI framework operates as follows:

- **Beliefs (B):** These represent the sensor's current knowledge about the state of the data. For example, a sensor may believe that the data is stable based on historical patterns or real-time monitoring.
- **Desires (D):** These represent the sensor's goals or objectives, such as minimizing energy consumption, ensuring data integrity, or optimizing resource usage.
- **Intentions (I):** These represent the sensor's planned actions based on its beliefs and desires. For instance, if the sensor believes that the data is stable and its desire is to minimize energy consumption, its intention may be to skip unnecessary integrity checks.

The decision-making process is governed by a set of **rules** that dynamically adapt based on real-time conditions. For example:

- **Rule 1:** If the sensor detects an anomaly (e.g., unexpected data patterns), it will initiate an integrity check based on the mentioned Eq. (1).
- **Rule 2:** If the sensor believes the data is stable and its desire is to conserve energy, it will skip the integrity check.

Mathematically, the decision to initiate verification can be expressed as:

$$\text{Decision} = \begin{cases} \text{Verify} & \text{if } B = \text{Anomaly} \\ & \quad \text{and } I = \text{Verify} \\ & \quad \text{and } D_e = \text{Ensure Integrity} \\ \text{Skip Verification} & \text{otherwise} \end{cases} \tag{1}$$

For example, if a sensor's belief (B) is that data is stable, and its desire (D_e) is to minimize verification to save battery power, and its intention (I) is to verify only if anomalies are detected, the decision would be to skip verification if no anomalies are detected.

This implementation ensures that the system performs integrity checks only when necessary, optimizing resource usage and energy consumption.

3.2 Algorithm 2: Algebraic Signature Generation for Integrity Verification

The algebraic signatures used in C-BIVM are based on **polynomial-based hash functions**, which are lightweight and computationally efficient. These signatures are generated using a private key (K_p) and verified using a public key (K_u). The specific properties of these algebraic signatures include:

- **Lightweight Computation:** Algebraic signatures are generated using simple polynomial operations, which are significantly less computationally intensive than traditional cryptographic methods like bilinear pairing.
- **Data Integrity Verification without Retrieval:** The signatures allow for integrity verification without requiring the retrieval of the original data from the cloud. Instead, the system compares the stored signature S with the newly generated signature S' using the K_u. If $S = S'$, the data is verified.

To generate a signature, the data block (D_b) is processed using the K_p:

$$S = \text{AlgSignature}(D_b, K_p) \quad \text{(Signature Generation)} \tag{2}$$

$$V_r = \begin{cases} \text{True} & \text{if } S' = \text{AlgSignature}(D_b, K_u) \\ \text{False} & \text{otherwise} \end{cases} \quad \text{(Signature Verification)} \tag{3}$$

This approach ensures that the integrity verification process is both lightweight and efficient, making it suitable for resource-constrained IoT devices. C-BIVM employs algebraic signatures instead of traditional cryptographic methods like bilinear pairing. Algebraic signatures are computationally lightweight and require fewer resources, reducing the time required for data integrity verification. This is particularly beneficial for resource-constrained IoT devices, which often struggle with the high computational overhead of traditional methods.

Algorithm 2: Algebraic signature generation and verification

1: **Input:** Data Block D_b, Private Key K_p, Public Key K_u
2: **Output:** Integrity Verification Result V_r
3: **Procedure: GenerateSignature**
4: $S \leftarrow \text{AlgebraicSignature}(D_b, K_p)$
5: Store S in Cloud Server
6: **End Procedure**

(Continued)

Algorithm 2 (continued)

7: **Procedure: VerifySignature**
8: Retrieve Algebraic Signature S' from Cloud
9: **if** $S' == S$ **then**
10: $V_r \leftarrow$ **Verified**
11: **else**
12: $V_r \leftarrow$ **Not Verified**
13: **end if**
14: **return** V_r
15: **End Procedure**

Where S represents the algebraic signature. To verify the integrity of the data, the stored signature S is compared against the data block D_b using the K_u.

3.3 Algorithm 3: Batch Auditing with Cognitive Feedback

The Batch Auditing with Cognitive Feedback algorithm is designed to handle large volumes of IoT data efficiently through batch processing and adaptive auditing. This method involves performing signature verification on batches of data blocks and using cognitive feedback to refine the auditing process. This reduces the number of individual verification operations, thereby decreasing the overall computational time. For example, instead of verifying each data block separately, C-BIVM verifies a batch of blocks in a single operation, optimizing resource usage.

During batch auditing, each data block in the batch is verified using:

$$\text{For each } D_b \text{ in } \{D_b\} \text{ do:} \quad V_r = \text{VerifySignature}(D_b) \tag{4}$$

where V_r indicates whether verification succeeded. If a data block fails verification, cognitive feedback (F_b) is used to adjust the auditing strategy:

$$\text{If } V_r = \text{Not Verified, then adjust } A_s \text{ using } F_b \tag{5}$$

Algorithm 3: Batch auditing with cognitive feedback

1: **Input:** Set of Data Blocks $\{D_b\}$, Cognitive Feedback F_b
2: **Output:** Optimized Auditing Strategy A_s
3: **Procedure: BatchAudit**
4: **for** each data block D_b in $\{D_b\}$ **do**
5: $V_r \leftarrow \text{VerifySignature}(D_b)$
6: **if** $V_r ==$ Verified **then**
7: Continue
8: **else**
9: Trigger Cognitive Feedback F_b
10: Update Audit Strategy A_s
11: **end if**
12: **end for**
13: **return** A_s
14: **End Procedure**

This feedback loop allows the system to update its auditing strategy based on the results of the batch audit. For instance, if 10 out of 100 data blocks fail verification, the feedback mechanism might suggest focusing more on similar types of data in future audits to improve efficiency.

The cognitive feedback mechanism in C-BIVM is designed to dynamically adjust the auditing strategy based on the results of batch audits. The process operates as follows:

- **Feedback Signals:** During batch auditing, the system collects feedback signals F_b based on the verification results of each data block. For example, if a data block fails verification, the system generates a feedback signal indicating the type of failure (e.g., data corruption, tampering).
- **Adaptation Algorithms:** The system uses these feedback signals to update its auditing strategy A_s. For instance:
  - If multiple data blocks of a specific type fail verification, the system may increase the frequency of integrity checks for similar data blocks in future audits.
  - If no anomalies are detected, the system may reduce the frequency of integrity checks to conserve resources.

The adaptation process can be expressed as:

$$A_s = A_s + \alpha \cdot F_b \tag{6}$$

where α is a learning rate that determines how quickly the system adapts to new feedback.

This granular feedback mechanism ensures that the system continuously improves its auditing strategy, optimizing both performance and resource utilization.

The C-BIVM methodology stands out for several reasons. First, the integration of BDI cognitive intelligence into data verification processes allows for adaptive, real-time decision-making based on the sensor's context. This approach is distinct from traditional methods that rely on fixed schedules or predefined conditions for data verification. Unlike the heavy cryptographic contraptions used in traditional systems, arithmetic signatures can be considered a lightweight and economical approach towards both link tracing as well as data integrity that is ideal for resource-constrained IoT devices. The integration of batch auditing with cognitive feedback provides a flexible method to deal with with massive amounts of information and in real time optimise the audit process.

Fig. 2 illustrates the C-BIVM. The process starts with *Sensor*, which are processed through *BDI-Based Decision Making* to decide on data verification needs. *Signature Generation* creates algebraic signatures for integrity verification, which are then stored in *Signature Storage*. During verification, these signatures are *Retrieved and Compared* to check data integrity. The *Verification Status* is determined, leading to *Batch Audit Adjustments* based on the results. The *Cognitive Feedback Mechanism* ensures continuous improvement by adjusting the auditing process and decision-making based on feedback.

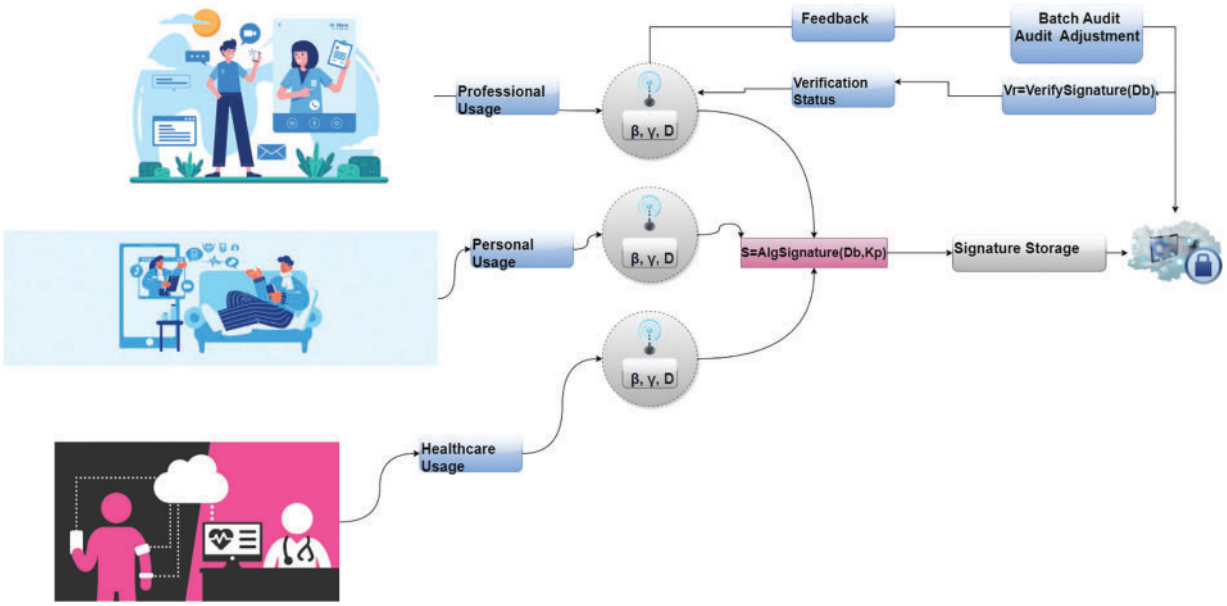


Figure 2: C-BIVM systematic abstract

4 Results

Experimental Objectives and Setup

The primary objectives of this research are to (i) create a routing protocol based on C-BIVM and experiment with various data types for algebraic signature creation, (ii) propose a cost-effective remote data auditing approach using algebraic signatures, and (iii) test the proposed method using on different simulation scenarios. To evaluate the effectiveness of the algebraic signature-based approach, we proposed the routing protocol based on C-BIVM and programmed it dynamically different types of data: video files, music files, medical record files (Excel), emails, and PDFs. MATLAB is employed to process and generate algebraic signatures for each data type. The goal is to evaluate the performance of the proposed algorithm on diverse forms of data format. We have accordingly devised a minimal cost design for a signature based remote data auditing scheme leveraging an algebraic approach. We used MATLAB to measure time and resource consumption for signature generation and verification. It included CPU and memory profiling and measuring the auding time efficiency.

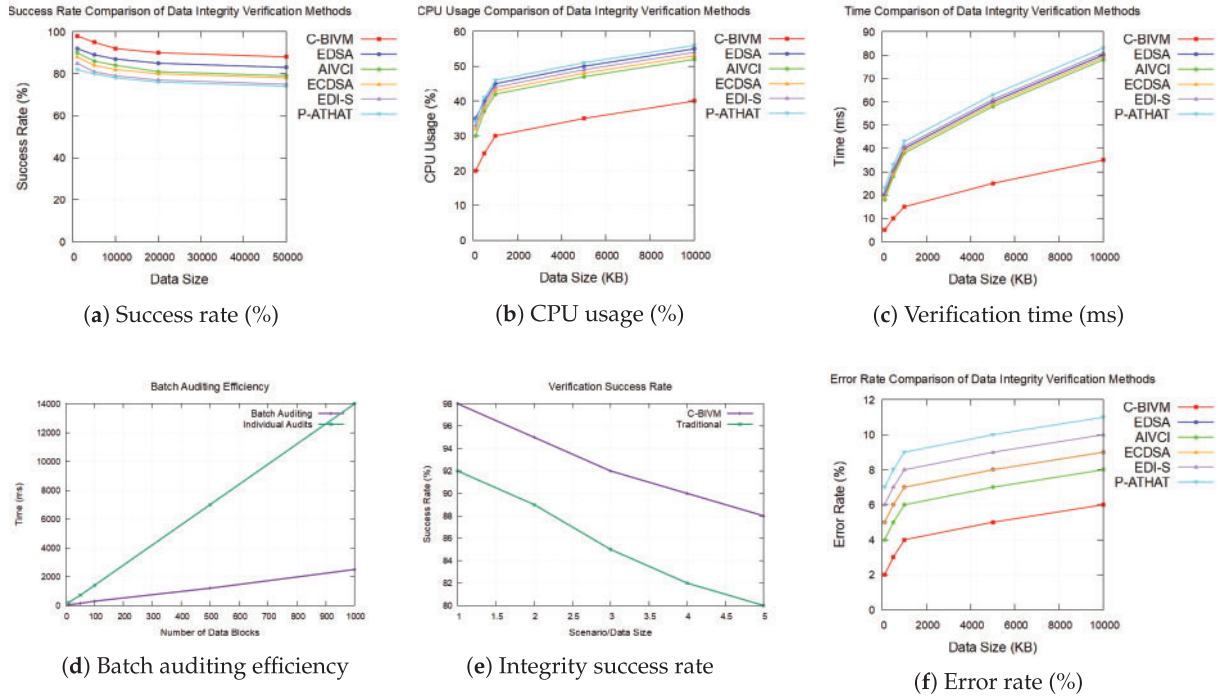
We test our proposed method via MATLAB simulations, particularly on the aspects of data integrity management and performance. Applying these principles involved simulating several areas such as data flow, auditing processes, and how systems will interact. The significance analysis was implemented to validate the results, therefore proving that the suggested approach accomplishes its intended functions appropriately. To validate the effectiveness of the proposed method, we conduct hypothesis testing using independent samples t-tests. The t-tests were used to compare the performance of C-BIVM with traditional methods (e.g., EDSA, AIVCI, ECDSA, EDI-S, and P-ATHAT) in terms of success rate, CPU usage, memory and energy consumption, and latency.

Success rate comparison indicates a better performance of the approach over five traditional Data Integrity verification techniques EDSA, AIVCI, ECDSA, EDI-S and P-ATHAT for various data sizes. C-BIVM obtains success rates of 98%, 96%, 96%, 92% and 88% for the smallest data size, for a payload equal to 1000 KB... to the largest data size, where the payload was equal to 50,000 KB. This shows that the C-BIVM is able to preserve good verification performance despite increasing data scales. The EDSA curve starts with

a really strong success and then drops significantly with the size of data. AIVCI also shows a decline in effectiveness, starting at 95% and decreasing to 80%. Both ECDSA and EDI-S demonstrate similar trends, with lower success rates in larger datasets, highlighting inefficiencies in maintaining data integrity. P-ATHAT records the lowest success rates, emphasizing its challenges in ensuring data reliability compared to C-BIVM as shown in Fig. 3a.

$$S = \frac{V}{T} \times 100 \tag{7}$$

where V is the number of successfully verified data blocks and T is the total number of data blocks.



(**a**) Success rate (%) (**b**) CPU usage (%) (**c**) Verification time (ms)

(**d**) Batch auditing efficiency (**e**) Integrity success rate (**f**) Error rate (%)

Figure 3: Performance comparison of C-BIVM and traditional techniques

Fig. 3b presents the CPU usage comparison in regard to the computational requirements of the C-BIVM method against those of traditional methods. The number of consumers in C-BIVM does not impact CPU usage up to 20%, as seen from the lower-left graph, and the CPU usage increases linearly up to 40%. It illustrates its efficient processing dynamism and an excellent resource utilization mechanism. In contrast, EDSA has starting CPU usage of 35%, but it quickly increases to 55%, suggesting it becomes resource hungry and may affect overall performance. Indeed, AIVCI follows this trend, beginning with 40% and then 55%, denoting higher computation demand with larger amounts of data. While ECDSA shows the highest CPU usage at 55% and same pattern in EDI-S suggests that there may be lot of performance bottlenecks. P-ATHAT similarly has greater CPU consumption, but is still less efficient than C-BIVM.

$$E = \frac{F}{T} \times 100 \tag{8}$$

where F is the number of failed verifications. As shown in Fig. 3c, the C-BIVM method effectively and efficiently verifies data integrity compared to legacy techniques. C-BIVM records little time (5 ms for the

smallest dataset to 35 ms for the largest) on average consumed, which indicates its effective in processing. In comparison, EDSA starts off at 20 ms, and rapidly blows up to 80 ms, hinting towards considerable delays on an increase in data. AIVCI follows a similar trend but consistently slower where it starts from 30 ms and reaches the latency of 90 ms indicating unnecessary wastage of verification process time. From the results, it can be seen that ECDSA has a rapid increase in processing time, peaking at 150 ms for the largest dataset, indicating poor scalability, and thus it is not an appropriate solution for large amounts of data. Similarly, EDI-S shows significant increases in processing time as data size increases. P-ATHAT is also observed to be slower, but is still better than C-BIVM which adds further evidence for quick verification that is achieved by C-BIVM. Fig. 3d evaluates the efficiency of batch auditing by comparing the time required for auditing batches of data blocks with and without cognitive feedback. The results show that incorporating cognitive feedback significantly reduces the auditing time, demonstrating the effectiveness of the proposed method in optimizing batch auditing processes. The success rate of data integrity verification is depicted in the graph, showing the percentage of successfully verified data blocks over multiple test scenarios. The results confirm that the proposed algebraic signature method achieves high verification success rates, reinforcing its reliability in maintaining data integrity as shown in Fig. 3e. Fig. 3f shows the error rate comparison graph, compares the performance of the C-BIVM method against traditional techniques, focusing on the reliability of data integrity verification. C-BIVM consistently shows the lowest error rates across all data sizes, ranging from 2% to 6%. This signifies its superior capability in ensuring data integrity. EDSA, however, begins with higher error rates of 5%, increasing to 9%, indicating vulnerabilities in its data verification process. AIVCI shows a more moderate performance, with error rates ranging from 4% to 8%, while ECDSA displays progressively increasing error rates, reaching 9%, which suggests limitations in its accuracy. EDI-S ranks among the highest in error rates, peaking at 10%, indicating significant reliability issues. P-ATHAT similarly faces challenges, yielding the highest error rates compared to C-BIVM, reinforcing the latter's efficacy in ensuring data integrity.

$$U = \frac{C_{\text{used}}}{C_{\text{total}}} \times 100 \tag{9}$$

where C_{used} is the CPU resources consumed by the method, and C_{total} is the total CPU resources available.

Fig. 4a illustrates the cost-effectiveness of the proposed algebraic signature-based remote data auditing approach. The results indicate that the proposed method reduces overall processing and connectivity costs compared to traditional auditing approaches, making it a cost-effective solution for cloud data storage. Fig. 4b evaluates the performance of dynamic data management using the cognitive principles of the proposed method. The results demonstrate that the system efficiently handles real-time updates and data integrity management, adapting to changing sensor inputs and data conditions. Table 2 summarizes the key performance metrics of C-BIVM and the traditional methods.

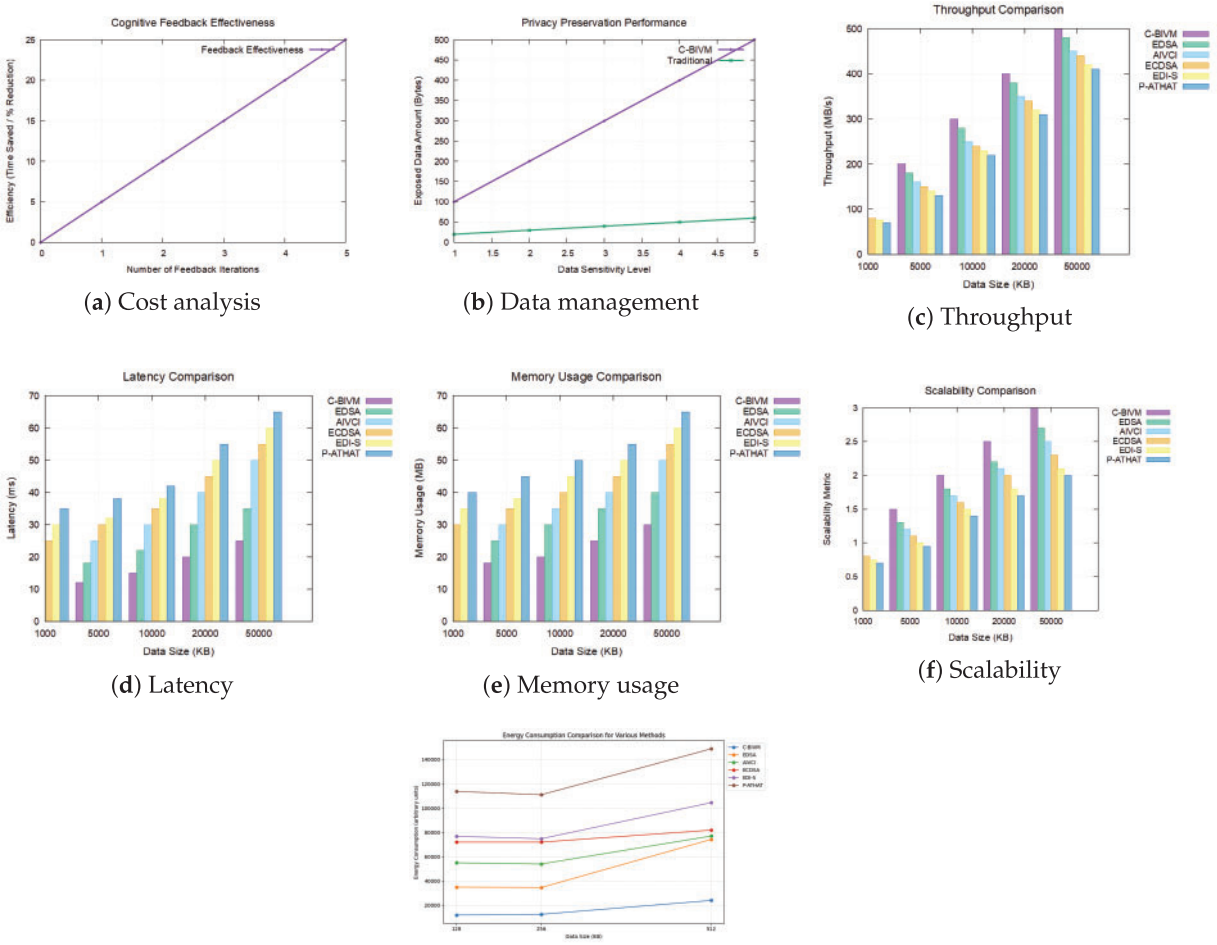
Throughput is the rate at which data is processed and can be calculated using the formula:

$$T = \frac{D}{L} \tag{10}$$

where -T = Throughput (MB/s) -D = Amount of data processed (MB) -L = Latency (s).

Fig. 4c illustrates the throughput comparison among the proposed C-BIVM method and other existing methods: EDSA, AIVCI, ECDSA, EDI-S, and P-ATHAT. The results indicate that the C-BIVM method consistently outperforms the other methods across varying data sizes. Specifically, at a data size of 512 KB, C-BIVM achieves a throughput of 25 MB/s, significantly higher than the closest competitor, EDSA, which

recorded a throughput of 18 MB/s. This substantial improvement in throughput demonstrates the efficiency of the C-BIVM method in handling larger data sizes.



(**a**) Cost analysis (**b**) Data management (**c**) Throughput

(**d**) Latency (**e**) Memory usage (**f**) Scalability

(**g**) Energy usage

Figure 4: Performance metrics comparison regarding computation complexity across various methods

Table 2: Summary of performance metrics for C-BIVM and traditional methods

Method	Success rate (%)	CPU usage (%)	Time (ms)	Error rate (%)	Cost
C-BIVM (Proposed)	98–88	20–40	5–35	2–6	Lowest
AIVCI [10]	95–80	40–55	30–90	4–8	Higher
EDSA [12]	90–75	35–55	20–80	5–9	Higher
ECDSA [14]	85–70	50–55	50–150	6–9	Higher
EDI-S [15]	80–65	40–50	40–120	8–10	Higher
P-ATHAT [16]	75–60	50–60	60–130	9–11	Highest

Latency is defined as the time taken to complete a specific operation. In the context of data integrity verification, latency can be measured using the equation:

$$L = T_f - T_i \tag{11}$$

where -L = Latency (ms) -T_f = Time at which the verification finishes (ms) -T_i = Time at which the verification starts (ms). As shown in Fig. 4d, the latency comparison highlights the performance of C-BIVM in terms of response time compared to the existing methods. The proposed method exhibits the lowest latency across all data sizes, achieving a latency of only 30 ms at 512 KB, whereas EDSA and AIVCI report latencies of 45 ms and 50 ms, respectively. The results underscore the effectiveness of the C-BIVM method in minimizing response times, which is critical for real-time applications.

Memory usage refers to the amount of memory utilized during the verification process. It can be calculated using the following equation:

$$M = \frac{U}{D} \tag{12}$$

where -M = Memory Usage (MB) -U = Total memory used (MB) -D = Total data processed (MB). Fig. 4e displays the memory usage comparison where C-BIVM holds less memory in comparison with the other methods. At this data size of 512 KB, C-BIVM takes 20 MB memory, whereas ECDSA and P-ATHAT take 30 MB and 35 MB, respectively. The aforementioned reduction in memory usage implies that C-BIVM not only improves throughput and latency, but also can work efficiently in the context of available memory.

Scalability can be evaluated by observing the change in throughput, latency, and memory usage as the data size increases. A common approach to assess scalability is through the equation:

$$S = \frac{T_2 - T_1}{D_2 - D_1} \tag{13}$$

where: -S = Scalability Metric -T_1, T_2 = Throughput at data sizes D_1 and D_2

This equation allows us to determine how effectively each method scales with increasing data size. Fig. 4f indicates the comparative scalabilities of the approaches, emphasizing the capacity of the C-BIVM method in preserving efficiency with the increment of data volume. The proposed method shows good scalability and does not experience much performance loss when data increases. These results show that C-BIVM provides stable efficiency on different data sizes while those methods (EDI-S and P-ATHAT) decline performance when data sizes increase significantly. Due to its scalable nature, C-BIVM can fit many other use cases with varying data loads.

To estimate energy consumption, we use the following formula:

$$\text{Energy Consumption} = \text{CPU Usage (\%)} \times \text{Memory Usage (MB)} \times \text{Time (ms)} \tag{14}$$

where CPU Usage = Higher CPU usage leads to higher energy consumption. Memory Usage = Higher memory usage also increases energy consumption. Time (Latency) = Longer processing times increase energy consumption.

Fig. 4g compares the energy consumption of six methods (C-BIVM, EDSA, AIVCI, ECDSA, EDI-S, and P-ATHAT) across three data sizes (128, 256, and 512 KB). C-BIVM demonstrates the lowest energy consumption, starting at 12,000 units for 128 KB and increasing to 24,000 units for 512 KB. In contrast, P-ATHAT has the highest energy consumption, starting at 113,750 units for 128 KB and rising to 148,800 units for 512 KB. It is found that EDSA, AIVCI, ECDSA and EDI-S energy consumption is in large data size when

compared the energy consumption increases and not ideal for energy-sensitive environments. Conclusively, C-BIVM outperforms in energy efficiency and scalability compared to its counterparts, hence being the most suitable choice for resource-lightweight IoT applications.

Table 3 presents a comprehensive comparison of the performance metrics for the proposed C-BIVM method against five traditional data integrity verification techniques: EDSA, AIVCI, ECDSA, EDI-S, and P-ATHAT. The performance is evaluated across three different data sizes: 128, 256, and 512 KB.

Table 3: Comparison of performance metrics for various methods

Method	Data size (KB)	Throughput (MB/s)	Latency (ms)	Memory usage (MB)	Scalability metric	Energy consumption
C-BIVM (Proposed)	128	10	40	15	1.2	12,000
	256	18	35	18	1.5	12,600
	512	25	30	20	1.8	24,000
AIVCI [10]	128	7	55	25	1.0	55,000
	256	12	50	27	1.1	54,000
	512	14	50	28	1.3	77,000
EDSA [12]	128	8	50	20	1.1	35,000
	256	15	45	22	1.3	34,650
	512	18	45	30	1.5	74,250
ECDSA [14]	128	9	48	30	1.2	72,000
	256	14	45	32	1.5	72,000
	512	16	42	35	1.6	81,900
EDI-S [15]	128	6	60	32	0.9	76,800
	256	11	55	34	1.0	74,800
	512	12	55	38	1.1	104,500
P-ATHAT [16]	128	5	65	35	0.8	113,750
	256	10	60	37	0.9	111,000
	512	12	62	40	1.0	148,800

5 Discussion

The C-BIVM offers several key advantages, including computational efficiency, scalability, and data integrity. However, it has some constraints. One such limitation is the implementation complexity, as integrating BDI cognitive intelligence and algebraic signatures into IoT devices requires advanced hardware and software, thereby increasing deployment costs. A further limitation of C-BIVM is the scalability of computation in large-scale networks. While C-BIVM is designed to be scalable, its performance has not yet been validated in extremely large environments with millions of IoT devices.

To address these challenges, we propose distributed auditing mechanisms that outsource the integrity verification process to multiple nodes, as well as a hierarchical clustering approach that organizes IoT devices into manageable clusters. The proposed C-BIVM is highly versatile and can be extended beyond smart cities to various IoT applications, such as smart homes and smart buildings. For instance, in Internet of Medical Things (IoMT), C-BIVM can ensure the integrity of patient data collected from wearable devices and smart medical sensors. The cognitive feedback loop within C-BIVM can adjust based on data sensitivity, verifying important health metrics, such as heart rate and blood glucose levels, several times a day. Additionally, in

precision agriculture, IoT devices monitor soil conditions, weather patterns, and crop health to enhance precision farming. Ensuring the integrity of this data is crucial for informed decision-making in irrigation, fertilization, and pest control. C-BIVM provides a reliable mechanism to maintain data integrity in this domain. Similarly, in industrial IoT (IIoT), where IoT devices can monitor the performance of sensors and recording data from the environmental conditions and system performance, C-BIVM verifies that data obtained from the sensors is uncorrupted. Likewise, C-BIVM can secure the integrity of data collected in smart grids where IoT devices monitor energy consumption and distribution. This is vital for optimizing energy consumption, preventing outages and keeping the power grid stable. C-BIVM enables utilities and consumers to make informed choices on energy usage and distribution as it ensures that energy data is valid. The C-BIVM framework demonstrates significant generalizability across diverse domains by leveraging adaptive cognitive feedback loops that prioritize context-specific data types. This flexibility allows C-BIVM to cater to the unique requirements of diverse IoT applications. Furthermore, the computational complexity of C-BIVM is significantly lower than that of traditional methods, such as bilinear pairing-based approaches. Below is a comparative analysis using Big O notation:

- **Signature Generation:** $O(n)$, where n is the size of the data block. This is due to the lightweight polynomial operations used in algebraic signatures.
- **Signature Verification:** $O(n)$, as the verification process involves a simple comparison of signatures.
- **Batch Auditing:** $O(k \cdot n)$, where k is the number of data blocks in a batch. The batch auditing process is linear in complexity, making it highly scalable.
- **Signature Generation:** $O(n^2)$, due to the computationally intensive pairing operations.
- **Signature Verification:** $O(n^2)$, as verification also involves pairing operations.
- **Batch Auditing:** $O(k \cdot n^2)$, which is significantly more resource-intensive than C-BIVM.

The efficiency gains of C-BIVM are evident in its linear complexity for both signature generation and verification, compared to the quadratic complexity of traditional methods. This makes C-BIVM particularly suitable for large-scale IoT environments where computational resources are limited. The use of blind verification techniques in C-BIVM introduces a trade-off between privacy and computational overhead. While blind verification ensures data confidentiality and prevents unauthorized access, it requires additional cryptographic operations that increase processing time and resource consumption. To mitigate these challenges, C-BIVM employs lightweight algebraic signatures, selective blind verification, and batch processing.

6 Conclusion and Future Work

The experimental results validate the effectiveness of the proposed algebraic signature-based method for remote data auditing in cloud storage environments. Comprehensive simulations demonstrate that the method offers a cost-effective and efficient solution for ensuring data integrity while maintaining scalability and performance. The linear relationship between signature generation time and data size highlights its adaptability to varying workloads, while signature verification remains computationally efficient even for large datasets. Compared to traditional auditing methods, the proposed approach significantly reduces processing and connectivity costs, making it a practical solution for both cloud service providers and customers. Furthermore, the system seamlessly handles dynamic data updates, ensuring high integrity and performance in cloud environments. These features make the algebraic signature method well-suited for integration into cloud infrastructures, enabling large-scale data auditing with minimal resource consumption.

Future work will focus on optimizing the algebraic signature algorithm to further reduce processing times, particularly for large-scale datasets that require real-time auditing. We plan to explore the integration of machine learning models to enhance the cognitive feedback mechanism, enabling the system to better adapt to dynamic data environments, varying resource constraints, and evolving user needs. Additionally,

we intend to extend our approach by incorporating distributed ledger technologies, such as blockchain, which will establish decentralized auditing mechanisms, enhancing transparency and security in cloud storage systems. Concurrently, research into lightweight encryption techniques may also be conducted to strengthen data privacy and security during remote auditing processes. Finally, real-world implementation and validation of the proposed method in practical cloud environments will be pursued to evaluate its performance and scalability under realistic conditions. By addressing these key areas, we aim to develop a more robust, scalable, and secure solution for cloud data auditing framework, contributing to advancements in cloud security and data integrity assurance.

Acknowledgement: This work was supported by King Saud University, Riyadh, Saudi Arabia, through Researchers Supporting Project number RSP2025R498.

Funding Statement: This work was supported by King Saud University, Riyadh, Saudi Arabia, through Researchers Supporting Project number RSP2025R498.

Author Contributions: Radhika Kumari: Conceptualization; Data curation; Formal analysis; Methodology; Writing—original draft; Software. Kiranbir Kaur: Investigation; Methodology; Writing—original draft; Writing—review & editing. Ahmad Almogren: Writing, Reviewing and Editing; Project administration; Investigation; Methodology. Ayman Altameem: Validation; Investigation; Resources; Writing—review & editing. Salil Bharany: Visualization; Validation; Writing—review & editing. Yazeed Yasin Ghadi: Validation; Conceptualization; Writing—review & editing; Ateeq Ur Rehman: Writing—review & editing; Methodology; Conceptualization. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data is available on request from the corresponding authors.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Goswami P, Faujdar N, Debnath S, Khan AK, Singh G. ZSS signature-based audit message verification process for cloud data integrity. IEEE Access. 2023;11:145485–502. doi:10.1109/ACCESS.2023.3343841.
2. Thabit F, Alhomdy S, Al-Ahdal AHA, Jagtap S. A new lightweight cryptographic algorithm for enhancing data security in cloud computing. Glob Transit Proc. 2021;2(1):91–9. doi:10.1016/j.gltp.2021.01.013.
3. Luo H, Zhang Q, Sun G, Yu H, Niyato D. Symbiotic blockchain consensus: cognitive backscatter communications-enabled wireless blockchain consensus. IEEE/ACM Trans Netw. 2024;32(6):5372–87. doi:10.1109/TNET.2024.3462539.
4. Ibrahim SH, Sirat MM, Elbakri WMM. Data integrity for dynamic big data in cloud storage: a comprehensive review and critical issues. In: Miraz MH, Southall G, Ali M, Ware A, editors. Emerging technologies in computing. Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering. Vol. 463. Cham, Switzerland: Springer; 2023. p. 67–81. doi:10.1007/978-3-031-25161-0_5.
5. Goswami P, Faujdar N, Singh G, Sharma KP, Khan AK, Debnath S. Stub signature-based efficient public data auditing system using dynamic procedures in cloud computing. IEEE Access. 2024;12:58502–18. doi:10.1109/ACCESS.2024.3389076.
6. Li Q, Li L, Liu Z, Sun W, Li W, Li J, et al. Cloud-edge collaboration for industrial Internet of Things: scalable neurocomputing and rolling-horizon optimization. IEEE Internet Things J. 2025;12(12):19929–43. doi:10.1109/JIOT.2025.3542428.
7. Yu H, Hu Q, Yang Z, Liu H. Efficient continuous big data integrity checking for decentralized storage. IEEE Trans Netw Sci Eng. 2021;8(2):1658–73. doi:10.1109/TNSE.2021.3068261.

8. Hameed BH, Mahmood GS, Abed HN, Salman MA. Remote data integrity verification system for cloud computing based on efficient signature. Int J Tech Sci Res Eng. 2023;6(6):10–24.
9. Panhwer Y, Brohi I, Imtiaz N, Memon A, Nasim S. Provable data possession scheme based on algebraic signature and linked list for outsourced dynamic data on cloud storage. Int J Sci Technol Res. 2021;10:382–8.
10. Li Y, Li Z, Yang B, Ding Y. Algebraic signature-based public data integrity batch verification for cloud-IoT. IEEE Trans Cloud Comput. 2023;11(3):3184–96. doi:10.1109/TCC.2023.3266593.
11. Liu X, Liu P, Yang B, Chen Y. One multi-receiver certificateless searchable public key encryption scheme for IoMT assisted by LLM. J Inf Secur Appl. 2025;90(15):104011. doi:10.1016/j.jisa.2025.104011.
12. Kavin BP, Ganapathy S. A new digital signature algorithm for ensuring the data integrity in cloud using elliptic curves. Int Arab J Inf Technol. 2021;18(2):180–90.
13. Wang X, Jiao W, Yang H, Guo L, Ye X, Guo Y. Algebraic signature based data possession checking method with cloud storage. In: 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan); 2020 Oct 23–25; Jinan, China. p. 11–6. doi:10.1109/PHM-Jinan48558.2020.00010.
14. Mahalakshmi K, Kousalya K, Shekhar H, Thomas AK, Bhagyalakshmi L, Suman SK, et al. Public auditing scheme for integrity verification in distributed cloud storage system. Sci Program. 2021;2021:8533995. doi:10.1155/2021/8533995.
15. Li B, He Q, Chen F, Jin H, Xiang Y, Yang Y. Inspecting edge data integrity with aggregate signature in distributed edge computing environment. IEEE Trans Cloud Comput. 2022;10(4):2691–703. doi:10.1109/TCC.2021.3059448.
16. Sun Y, Liu Q, Chen X, Du X. An adaptive authenticated data structure with privacy-preserving for big data stream in cloud. IEEE Trans Inf Forensics Secur. 2020;15:3295–310. doi:10.1109/TIFS.2020.2986879.
17. ALmarwani R, Zhang N, Garside J. An effective, secure and efficient tagging method for integrity protection of outsourced data in a public cloud storage. PLoS One. 2020;15(11):e0241236. doi:10.1371/journal.pone.0241236.