Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059949

ARTICLE

Provable Data Possession with Outsourced Tag Generation for AI-Driven E-Commerce

Yi Li[1], **Wenying Zheng**[2], **Yu-Sheng Su**[3,4,5,*] **and Meiqin Tang**[6]

[1]School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, 210044, China
[2]School of Computer Science and Technology (School of Artificial Intelligence), Zhejiang Sci-Tech University, Hangzhou, 310018, China
[3]Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, 621301, Taiwan
[4]Advanced Institute of Manufacturing with High-tech Innovations, National Chung Cheng University, Chiayi, 621301, Taiwan
[5]Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, 202301, Taiwan
[6]School of Integrated Circuits, Wuxi Vocational College of Science and Technology, Wuxi, 214028, China

*Corresponding Author: Yu-Sheng Su. Email: ccucssu@gmail.com

Received: 21 October 2024; Accepted: 25 February 2025; Published: 16 April 2025

ABSTRACT: AI applications have become ubiquitous, bringing significant convenience to various industries. In e-commerce, AI can enhance product recommendations for individuals and provide businesses with more accurate predictions for market strategy development. However, if the data used for AI applications is damaged or lost, it will inevitably affect the effectiveness of these AI applications. Therefore, it is essential to verify the integrity of e-commerce data. Although existing Provable Data Possession (PDP) protocols can verify the integrity of cloud data, they are not suitable for e-commerce scenarios due to the limited computational capabilities of edge servers, which cannot handle the high computational overhead of generating homomorphic verification tags in PDP. To address this issue, we propose PDP with Outsourced Tag Generation for AI-driven e-commerce, which outsources the computation of homomorphic verification tags to cloud servers while introducing a lightweight verification method to ensure that the tags match the uploaded data. Additionally, the proposed scheme supports dynamic operations such as adding, deleting, and modifying data, enhancing its practicality. Finally, experiments show that the additional computational overhead introduced by outsourcing homomorphic verification tags is acceptable compared to the original PDP.

KEYWORDS: Provable data possession; data auditing; cloud computing; e-commerce; bloom filter

1 Introduction

AI is Omnipresent [1,2]. AI is present in voice assistants and image recognition on smartphones and devices, as well as in recommendation systems. It also plays a role in personalized content recommendations, advertising, and fraud detection on social media. Additionally, AI has widespread applications in specific industries and enterprises, such as healthcare, finance, manufacturing, and transportation [3–7]. In summary, the presence and impact of AI are ubiquitous.

In the e-commerce sector, the demand for AI is even greater. E-commerce platforms can easily obtain vast amounts of business data, which, when analyzed, can power a wide range of AI applications, such as personalized product recommendations, targeted advertising, market forecasting, and decision-making [8,9]. These AI applications rely on high-quality, accurate data; if the data is incomplete or incorrect,

it will inevitably affect the effectiveness of the AI applications [10]. Therefore, it is crucial to verify the integrity of the data before deploying AI applications in e-commerce [11].

Provable Data Possession (PDP) is a protocol used to verify the integrity of cloud data. PDP works by dividing the data uploaded to the cloud server into smaller data blocks, then generating homomorphic verification tags, and subsequently verifying the integrity of the cloud data through sampling queries [12,13]. However, PDP is not suitable for e-commerce due to the significant computational resources required to generate the homomorphic verification tags, especially given the large volume of data in e-commerce [14]. To ensure the integrity of the data supporting AI services is not tampered with while reducing the computational overhead of the e-commerce server, we propose PDP with Outsourced Tag Generation (OTGPDP) for AI-driven e-commerce, which outsources the generation of homomorphic verification tags to the cloud server and incorporates a lightweight tag verification method based on Counting Bloom Filter (CBF) to prevent malicious tag generation [15]. The main contributions are as follows:

- Building on PDP, a lightweight verification method for homomorphic verification tags was constructed using CBF. This ensures that the homomorphic verification tags generated by the cloud server match the uploaded data.
- The proposed OTGPDP supports dynamic data operations. When data is added, deleted, or modified, only the CBF corresponding to the challenged data blocks needs to be updated, thereby reducing the computational overhead of dynamic operations.
- Finally, the experimental results indicate that the additional computational overhead introduced by outsourcing the generation of homomorphic verification tags in OTGPDP is acceptable.

2 Literature Review

This section first reviews the research on the integration of e-commerce and AI and then summarizes the relevant studies on PDP.

2.1 AI-Driven E-Commerce

The integration of AI and e-commerce has become a hot topic, whether it is in helping e-commerce improve automation efficiency or optimizing the accuracy of recommendation systems within e-commerce.

Verma et al. proposed a data-driven logistics modeling framework to address the challenges of logistics management in e-commerce [16]. This framework integrates descriptive, predictive, and prescriptive analytics to provide e-commerce businesses with more comprehensive decision-making references. It enables more accurate forecasting of sales and shipping costs, optimizes the layout of logistics facilities, and significantly enhances the efficiency of logistics systems.

Barata et al. introduced a multi-criteria decision analysis model to help small and medium-sized enterprises integrate e-commerce and AI resources to enhance their competitiveness [17]. This model identifies and analyzes the e-commerce and AI methods that are advantageous or disadvantageous for small and medium enterprises, providing decision support to clarify the factors that need to be prioritized and strengthened, thereby improving overall business performance.

Ogbeyemi et al. introduced a human factors analysis-based design method for e-commerce delivery networks to tackle the issue of delivery delays caused by supply chain disruptions [18]. This method analyzes human factors in warehouses and delivery systems, such as skill levels, fatigue, and work rotation, using exploratory data analysis and statistical modeling to reveal potential impacts. It offers a new perspective for researching highly reliable e-commerce delivery networks.

Yu et al. proposed a new enterprise semantic model to address the challenge of effectively integrating complex data formats in enterprise application integration [19]. This model separates business functions from application systems, adopting a three-layer architecture that supports the integration of structured, semi-structured, and unstructured data, thereby enhancing the flexibility and resilience of enterprise systems.

Wang et al. proposed a technology acceptance model for AI-driven e-commerce to address the issue of understanding user acceptance and usage of AI technologies in e-commerce [20]. This model captures users' behavioral intentions and subjective attitudes toward actual AI usage, and it uses a partial least squares regression model to analyze the data, validating the applicability of the technology acceptance model in the context of e-commerce.

Upadhyay et al. developed a systematic data-driven benchmark marketing approach to overcome the shortcomings of data-driven precision marketing methods, which often fail to fully utilize diverse data sources and lack an effective understanding of customer behavior [21]. This approach uses various machine learning algorithms to predict and classify customer behavior, enabling highly targeted marketing strategies. Experimental results demonstrate that this method significantly improves the efficiency of marketing activities and increases the return on investment for businesses.

Tsang et al. proposed a supply chain performance analysis system based on blockchain sharding technology to address the deficiencies in supply chain performance measurement models in the context of Industry 5.0 [22]. This system divides the blockchain network into multiple shards and employs an adaptive fuzzy inference system for accurate and reliable measurement. It enhances data management and performance measurement within the supply chain, providing greater flexibility and adaptability in high-load supply chain environments.

2.2 PDP

Shen et al. proposed a certificateless PDP scheme addressing the limitations of cloud-based electronic health data integrity in handling dynamic operations and data tracking [23]. This scheme stores multiple copies of electronic health data across multiple cloud servers and designs block-level dynamic operations. It also introduces a new dynamic data structure that resists replica aggregation attacks, avoiding the certificate management burden found in traditional PDP schemes and improving overall system efficiency.

Yang et al. tackled the high computational costs associated with complex bilinear pairing operations in traditional identity-based PDP schemes by proposing a more efficient identity-based PDP protocol [24]. This scheme uses only basic algebraic operations and introduces a compressed cloud storage technique. Additionally, it can be extended to support user revocation, dynamic data updates, and batch auditing.

Wang et al. addressed the single point of failure issue caused by the centralized key management in traditional PDP schemes by proposing a PDP platform named SStore, which implements decentralized key management and damaged data block localization [25]. The platform also uses basic algebraic operations to reduce computational overhead. Validation on an actual integrated platform demonstrated that SStore enhances system robustness.

Li et al. proposed a blockchain-based synchronous PDP scheme to address the lack of effective time-state verification mechanisms in traditional PDP in digital twin environments [26]. This scheme uses blockchain to provide a time-state synchronization window, ensuring that tags and data blocks within the same time period can be aggregated and verified for integrity. Additionally, the scheme supports flexible verifier selection and uses blockchain's anonymity services to protect the identities of entities during verification.

Deng et al. introduced a new certificateless PDP scheme that, unlike existing schemes that rely on the random oracle model for security, is proven secure under the standard model [27]. By reducing the number

of hash-to-point operations, the scheme ensures that the number of bilinear pairing operations is limited to three, thereby reducing computational overhead.

Wang et al. proposed a blockchain-based synchronous distributed PDP scheme to address the low computational efficiency and lack of forward security in traditional PDP [28]. This scheme incorporates blockchain technology to achieve time synchronization and data immutability, ensuring forward security in data integrity verification. The authors also formalized the system model and security model for this problem.

Dhakad et al. proposed an efficient privacy-preserving PDP scheme to address the shortcomings of traditional PDP in user privacy and dynamic operations [29]. This scheme uses bilinear pairing and hash functions to design the integrity verification structure and supports batch auditing and dynamic operations. Furthermore, it avoids direct transmission of raw data during the auditing process, thereby protecting user privacy.

From the above studies, it is evident that PDP research has been a hot topic in recent years. However, none of the above schemes have addressed the outsourcing of audit metadata, which is crucial for reducing the computational overhead for data owners in generating audit metadata—this is the problem that this paper aims to solve.

2.3 Organization

The remainder of this paper is organized as follows: Section 3 provides an overview of the scheme model, followed by a detailed explanation of the proposed scheme in Section 4. Sections 5 and 6 thoroughly analyze the scheme's security and performance, respectively. Lastly, Section 8 concludes the paper with a summary and suggestions for future research directions.

3 Scheme Model

This section first illustrates the system model of OTGPDP and then its threats and security model.

3.1 System Model

OTGPDP contains only two entities: an e-commerce server and a cloud server which are depicted as follows:

E-commerce server: is the actual owner of the e-commerce data and seeks to leverage cloud servers to access AI services, such as providing data to obtain AI models. Its computational and storage resources are limited, so it needs to store data on a cloud server and periodically verify the integrity of the e-commerce data stored there. Additionally, it relies on the cloud server to generate homomorphic verification tags.

- *Cloud server:* is responsible for storing the data from the e-commerce server and providing computational services, including AI model training and the generation of homomorphic verification tags. It is malicious but rational, meaning that due to reputational concerns, it will avoid engaging in malicious activities that are likely to be detected.

Next, the framework of OTGPDP is summarized in Definition 1.

Definition 1: *OTGPDP contains eight algorithms:*

Setup: *Given a security parameter, the e-commerce server generates the system parameters and distributes them. In addition, e-commerce servers and cloud server have their public and private keys separately and share the public keys.*

- ***DataProcess:*** *Given the local e-commerce data, the e-commerce server first divides them into small data blocks. If the e-commerce server has confidentiality requirements for the data, it can apply a random mask or use symmetric encryption to protect the data's privacy. Finally, the data blocks are sent to the cloud server.*
- ***MetaGen:*** *Given the e-commerce data, the cloud server first generates the homomorphic verification tags and constructs the CBF for the tags. Then, the cloud server sends the tags and CBF to the e-commerce server.*

MetaVer: *Given the tags and CBF, the e-commerce server verifies them. If the verification, the e-commerce server stores the CBF and deletes other data including the e-commerce data.*

Challenge: *Given the audit period and the number of challenge data blocks, the e-commerce server generates two random number seeds and sends them to the cloud server.*

- ***Proof:*** *Given the random number of seeds and the number of challenge data blocks, the cloud server generates a proof based on the stored e-commerce data and returns it to the e-commerce server.*
- ***Verify:*** *Given the proof sent by the cloud server, the e-commerce server verifies it.*
- ***DataUp:*** *Given specific dynamic operation types, including addition, deletion, and modification, the cloud server modifies the local data and updates the CBF.*

3.2 System Model

Intuitively, OTGPDP faces the following threats. The cloud server may lose or tamper with the stored data due to system errors. Concerning its reputation, it might forge a CBF and proof that can pass verification. Additionally, even if only a small portion of the e-commerce data is damaged, the e-commerce server should still be able to detect this malicious behavior. Based on these threats, we propose the following security model, which is formalized via the following definitions.

Definition 2 (Correctness): *The OTGPDP protocol satisfies correctness if the cloud server and e-commerce server honestly follow the proposed procedures, ensuring that the CBF can pass the MetaVer algorithm and the integrity proof can pass the Verify algorithm.*

Definition 3 (Unforgeability): *The OTGPDP protocol achieves unforgeability if, in the event of e-commerce data damage, the cloud server can reconstruct the integrity proof to pass verification with a negligible probability.*

Definition 4 (Detectability): *The OTGPDP protocol achieves detectability if, when only a small fraction of the data on the cloud server is damaged, the e-commerce server can still identify this abnormality with a non-negligible probability.*

4 Methods

4.1 Overview

The main design concept of OTGPDP is to build upon the PDP framework by utilizing the cloud server to generate homomorphic verification tags. While the cloud server generates these tags, it also constructs a corresponding CBF and sends it to the e-commerce server. The e-commerce server then randomly selects a portion of the data blocks and generates the corresponding homomorphic verification tags to verify the CBF. If the verification is successful, the CBF is used for subsequent data auditing tasks. The specific protocol flow is shown in Fig. 1.

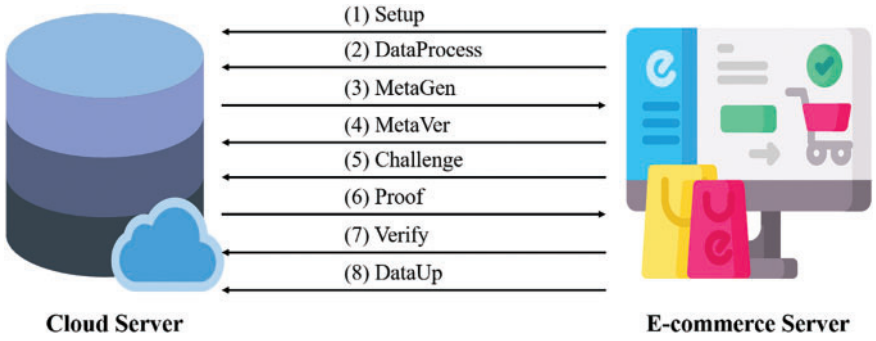

Figure 1: Overview of OTGPDP

4.2 Detailed OTGPDP

This section provides the detailed design of the eight algorithms in OTGPDP, as follows.

4.2.1 Setup (λ) → (SK, PK)

Let g be the generator of a group $\mathcal{G}$ with order p, where p is a large prime satisfying the security parameter λ. The e-commerce server defines a pseudo-random number generator PN(·) and a set of hash functions H(·). Using a random seed, PN(·) generates a sequence of random numbers {rn1, rn2, rn3, . . .}. The e-commerce server chooses a private key x from Z_p^* and calculates the public key $X = g^x$. Similarly, the cloud server generates its private key y and corresponding public key $Y = g^y$. The secret key set SK contains x, y, while the public key set PK consists of $\{\mathcal{G}, g, p, X, Y, PRNG(\cdot), H(\cdot)\}$.

4.2.2 DataProcess (D^*) → (D)

The e-commerce server first divides the commerce data into n data blocks, then selects a random seed to generate a random mask, which is used to create scrambled data that protects the contents of the commerce data. Alternatively, symmetric encryption algorithms can be used to achieve data privacy protection. Whether or not to protect the original data depends on the requirements of the e-commerce server.

- The e-commerce server divides D^* into n data blocks $\{d_1^*, d_2^*, \ldots, d_n^*\}$.
- The e-commerce server selects a random seed seed1 to generate the random masks, $\{rn_1^1, rn_2^1, \ldots, rn_n^1\}$.
- The e-commerce server generates the uploading data $D = \{d_1, d_2, \ldots, d_n\}$, in which, $d_i = d_i^* + rn_i^1$.
- The e-commerce server sends the D to the cloud server.

4.2.3 MetaGen (D, X) → (Tag, CBF)

After receiving the data and public key from the e-commerce server, the cloud server first generates the corresponding homomorphic verification tags. It then creates a counting bloom filter for these tags and returns it to the e-commerce server.

- The cloud server computes the homomorphic verification tags $Tag = \{t_1, t_2, \ldots, t_n\}$, in which, $t_i = H(filename||i) \times g^{d_i}$
- The cloud server calculates the bit array size by reversing the false positive probability formula of the Bloom filter. $\sigma = (1 - \left(1 - \frac{1}{m}\right)^{kn})^k$, where, σ is the false positive probability, m is the bit array size, k is the number of hash functions, and n is the number of data blocks. The $m = \frac{-k^2 n}{\ln(p)}$ [30–32].

- The cloud server calculates the hash value of t_i with k hash functions, $H^1(\cdot), H^2(\cdot), \ldots, H^k(\cdot)$.
- The cloud server increments the value by one at the corresponding positions with an index of the hash value in the bit array (*CBF*). Then, the cloud server sends the *CBF* to the e-commerce server.

4.2.4 MetaVer (CBF, D) → (True/Flase)

When the e-commerce server receives the CBF computed by the cloud server, it randomly selects a certain proportion of data blocks from the locally stored e-commerce data and calculates the corresponding homomorphic verification tags. It then verifies whether these tags are present in the CBF. If they are all present, the e-commerce server accepts the CBF and deletes the local data.

- The e-commerce server selects a portion of data blocks (c) and computes the homomorphic verification tags $Tag^s = \{t_{s_1}, t_{s_2}, \ldots, t_{s_c}\}$, in which, $t_{s_i} = H(filename||s_i) \times g^{d_{s_i}}$.
- The e-commerce server calculates the hash value of $t_s i$ with k hash functions, $H^1(\cdot), H^2(\cdot), \ldots, H^k(\cdot)$.
- The e-commerce server verifies whether the values at the k hash positions of each t_{s_i} in the *CBF* are greater than 1. If all values are greater than 1, the e-commerce server accepts the *CBF* and deletes the local e-commerce data.

4.2.5 Challenge (·) → (chal, chal^{sig})

The e-commerce server initiates a challenge by generating chal and its signature $chal^{sig}$, which are then sent to the cloud server.

- The e-commerce server first determines the number of data blocks to be audited, denoted by c.
- The e-commerce server generates a random value sd^2 to select the indices of the data blocks for auditing.
- Another random value, sd^3, is generated to compute the factors for these blocks. The challenge *chal* is then formed as the tuple c, sd^2, sd^3.
- Finally, the e-commerce server signs *chal* to produce $chal^{sig}$ and sends both *chal* and $chal^{sig}$ to the cloud server.

4.2.6 Proof (chal, chalsig) → (P, P^{sig}, Tag^c)

After receiving the *chal*, the cloud server generates a proof P based on the locally stored data and sends it to the e-commerce server.

- The cloud server first extracts the index $IND = \{PN\ (sd^2)_1, PN\ (sd^2)_2, \ldots, PN\ (sd^2)_c\}$ and the coefficients $FAC = \{a_1, a_2, \ldots, a_c\}$, where $a_i = PN\ (sd_i^3)$, based on the *chal*.
- The cloud server computes the $P = g \sum_{i \varepsilon IND} a_i \times m_{i.}$
- The cloud server picks the tags with the chosen index Tagc = $\{t_{c_1}, t_{c_2}, \ldots, t_{c_c}\}$.
- The cloud server generates the signature of P ^{sig} and sends P, P ^{sig}, Tag^c to the ecommerce server.

4.2.7 Verify (chal, P, P^{sig}, Tag^c) → (True/False)

After receiving *chal*, P, P^{sig}, and Tag^c, the e-commerce server first uses the *CBF* to verify all the challenge tags Tag^c. If the verification is successful, it then proceeds to verify the audit results.

- The cloud server first computes the k hash functions for all tags: $H^1(t_i), H^2(t_i), \ldots, H^k(t_i)$. It then verifies whether the corresponding positions in the *CBF* are all greater than 1. If the verification is successful, it proceeds to the next step.
- The cloud server computes $\delta = \prod_{i \varepsilon IND} Tag_i^{a_i}$.

- Blockchain Node verify $\delta \stackrel{?}{=} \prod_{i \epsilon IND} H(filename \parallel i)^{a_i} \times P$. If they are equal, return True; if not, return False.

4.2.8 DataUp (op) → (CBF^{update})

Regarding dynamic operations for data auditing, mature solutions already exist in the form of dynamic PDP protocols. The process for dynamic operations on the homomorphic verification tags' CBF is as follows:

add: $\{op = (ADD, n + 1), d^*_{update}\}$, e-commerce server first generates the $d_{updata} = d^*_{update} + PN\ (sd1)$ with index $n + 1$. Then, it updates the CBF by incrementing 1 with the position of $H^1(d_{updata})$, $H^2(d_{updata})$, …, $H^k(d_{updata})$. Finally, op, d_{updata}, CBF^{updata} are sent to a cloud server.

- delete: $\{op = (DEL, ind), d^*_{update} = \varnothing\}$, e-commerce server first generates the $d_{updata} = \varnothing + PN\left(sd^1\right)$ with index ind. Then, it updates the CBF by incrementing 1 with the position of $H^1(d_{updata})$, $H^2(d_{updata})$, …, $H^k(d_{updata})$. Finally, op, d_{updata}, CBF^{updata} are sent to a cloud server.
- modify: $\{op = (MOD, ind), d^*_{update}\}$, e-commerce server first generates the $d_{updata} = d^*_{update} + PN\ (sd^1)$ with index ind. Then, it updates the CBF by incrementing 1 with the position of $H^1(d_{updata})$, $H^2(d_{updata})$, …, $H^k(d_{updata})$. Finally, op, d_{updata}, CBF^{updata} are sent to a cloud server.

5 Security Analysis

5.1 Correctness

Theorem 1: *If the cloud server and e-commerce server execute OTGPDP honestly, the integrity proof generated by the cloud server can always pass the e-commerce server's verification.*

Proof: The integrity proof of the challenged blocks is P. The TPO has to verify whether $\prod_{i \epsilon IND} H(filename||i)^{a_i} \times P = \sigma$ holds. It is shown as follows:

$$\begin{aligned}
&\prod_{i \epsilon IND} H(filename||i)^{a_i} \times P \\
&= \prod_{i \epsilon IND} H(filename||i)^{a_i} \times g^{\sum_{i \epsilon IND} a_i \times m_i} \\
&= \prod_{i \epsilon IND} (H(filename||i)^{a_i} \times g^{a_i \times m_i}) \\
&= \prod_{i \epsilon IND} (H(filename||i) g^{m_i})^{a_i} \\
&= \prod_{i \epsilon IND} t_i^{a_i} \\
&= \sigma
\end{aligned}$$

Therefore, OTGPDP satisfies the property of correctness. □

5.2 Unforgebility

Theorem 2: *If the e-commerce data is damaged, the cloud server can reconstruct the auditing proof passing verification with a negligible probability.*

Proof: When data on the cloud server is corrupted, even though the cloud server might generate a correct homomorphic verification tag for the corrupted data, and that tag could potentially pass the CBF verification, it would require the cloud server to construct k hash values such that the corresponding positions in the CBF are greater than 1. Due to the collision resistance of hash functions, the cloud server can only achieve this through brute force, with a probability of $(\frac{1}{m})^k$, which is negligible. Additionally, CBF itself has a false positive probability, denoted as p, which can be set to a negligible level based on security requirements. Therefore,

the probability that the cloud server can reconstruct an auditing proof that passes verification is $(\frac{1}{m})^k$ + p, which is negligible. □

5.3 Detectability

Theorem 3: *if only a small fraction of the e-commerce data on the cloud server is damaged, the e-commerce server can still disclose this abnormality with a non-negligible probability.*

Proof: Assume an edge server removes k data blocks from a file of *n* total blocks. Let *c* be the number of blocks selected for verification. The probability of identifying the missing blocks, denoted as $P_{detected}$, is calculated as: $P_{\text{detected}} = 1 - P_{\text{undetected}} = 1 - \frac{n-k}{n} \cdot \frac{n-k-1}{n-1} \cdot \frac{n-k-2}{n-2} \cdot, \ldots, \cdot \frac{n-k-c+1}{n-c+1}$. Given that $\frac{n-i-k}{n-i} > \frac{n-i-1-k}{n-i-1}$, it follows that $1 - (\frac{n-k}{n})^c < P_{detected} < 1 - (\frac{n-k-c+1}{n-c+1})$. This probability is substantial. Intuitively, with a 1% loss, sampling 4.6% of data blocks yields a detection rate of over 99%. □

6 Results

We conducted a performance analysis of the additional computational overhead introduced by OTG-PDP using the Python programming language on a virtual machine. The security strength was set to 1024 bits, and the hash function used was the SHA256 algorithm with different random seeds. The number of hash functions was set to 3, and the false positive rate of the CBF was set to 0.001. The files used were randomly generated binary. bin files with sizes of 200, 400, 600, and 800 MB. The data block sizes were set to 2, 4, 8, 16, and 32 KB. The virtual machine was configured with 8 virtual CPUs and 16 GB of RAM. The physical machine's CPU model was Intel i9-13980HX. In addition, the latency information between nodes in cloud computing or edge computing is as follows: within the same data center, the latency is typically between 1 and 10 microseconds. For inter-data center communication within the same city, the latency is usually between 1 and 5 ms [32]. It can be seen that if deployment is limited to a single country, network communication latency is generally within 5 ms. Since e-commerce networks are typically deployed within a single country, the inter-node latency does not exceed 5 ms in the experiments. Since the computational overhead of the OTGPDP protocol is typically on the scale of seconds, we ignored the communication latency between nodes in the experiments.

6.1 Origin Data Generation

The time cost of generating the AI data used in the experiment is shown in Fig. 2.

As shown in Fig. 2, the initial data used for the subsequent experiments was generated using Python's built-in random function. On one hand, with the same data block size, the larger the file, the greater the time overhead. On the other hand, with the file size remaining constant, the larger the data block size, the lower the time overhead. The main reason is that larger data blocks require more time for the random function to generate, but the total number of data blocks decreases. This indicates that the number of data blocks is the dominant factor in file generation. Additionally, the time required to generate all files is less than 2 s.

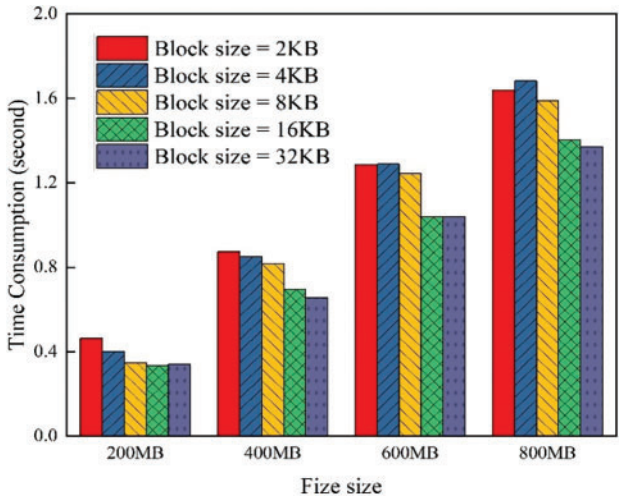

Figure 2: Origin data generation time with different file size and block size

6.2 Tag Generation

The tag generation time for files with different data block sizes is shown in Fig. 3.

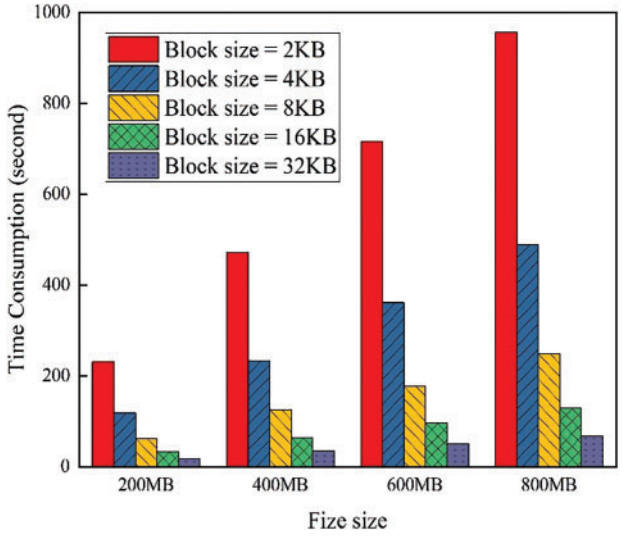

Figure 3: Tag generation time with different file sizes and block size

As seen in Fig. 3, it is clear that generating homomorphic verification tags is a time-consuming step. Additionally, when the data block size remains constant, the larger the file, the greater the time overhead, showing an overall linear increase because the number of data blocks increases linearly. On the other hand, when the file size remains constant, the time overhead exhibits an exponential change with variations in data block size. This is primarily because generating homomorphic verification tags involves calculations modulo the security parameter, leading to a similar time overhead for tag generation across different data blocks, resulting in an exponential decrease in time overhead as the block size increases.

6.3 CBF Generation

The time overhead and storage overhead for constructing the CBF are shown in Figs. 4 and 5, respectively. Fig. 6 shows the computational overhead of CBF verification.

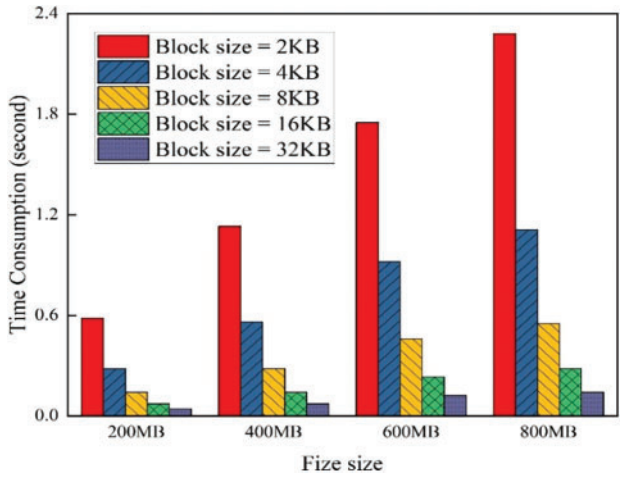

Figure 4: CBF construction time with different file size and block size

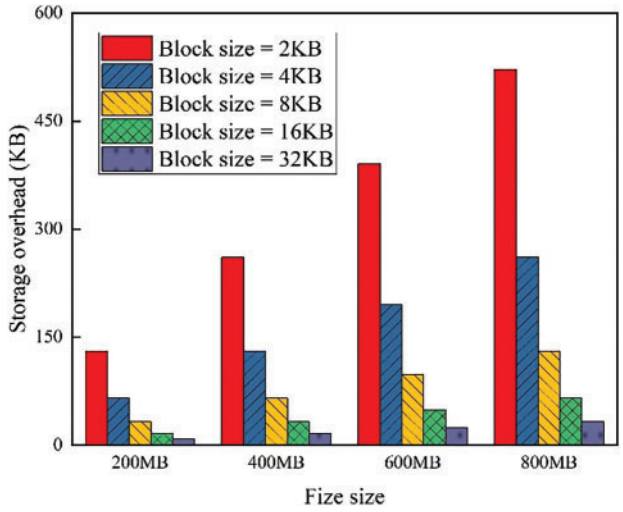

Figure 5: CBF storage overhead with different file size and block size

As shown in Fig. 4, the time overhead for constructing the CBF is relatively low. For an 800 MB file with 2 KB data blocks, the time overhead is just over 2 s, which does not introduce significant additional computational overhead. Similarly, when the data block size remains constant, the computational overhead increases linearly with the file size. Conversely, when the file size remains constant, the computational overhead generally exhibits an exponential decrease with increasing data block size. Notably, the rate of decrease is slightly less than half, because the CBF requires calculations modulo the bit array size; as the number of data blocks increases, the bit array size also increases, leading to some additional computational overhead.

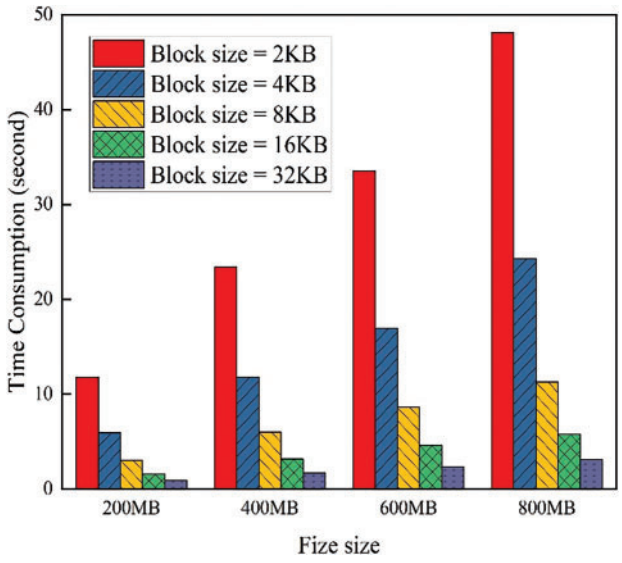

Figure 6: Computational overhead for CBF verification with different file size and block size

As shown in Fig. 5, the additional storage overhead brought by the CBF is only a few hundred kilobytes. Compared to traditional PDP, where homomorphic verification tags are often on the order of megabytes, OTGPDP does not impose additional storage overhead on the user. Moreover, the trend in storage overhead decreases exponentially with increasing data block size and increases linearly with file size. The underlying reason is that the combined effect of data block size and file size on the total number of data blocks remains consistent, while the impact on storage overhead for each individual data block does not change. In experiments conducted in this study, each element of the CBF's bit array is an 8-bit number.

As shown in Fig. 6, the computational overhead for CBF verification exhibits a trend similar to that in Fig. 3. The main reason is that the e-commerce server needs to randomly select 5% of the data blocks to compute homomorphic verification tags and verify whether they are in the CBF. Since the verification overhead of the CBF is very low, it hardly affects the overall time overhead trend. It is not difficult to observe that the computational overhead for CBF verification, with the same data block size and file size, is essentially 5% of that shown in Fig. 3.

6.4 Extra Computing Overhead of Audit

Since the audit process in OTGPDP only adds an additional step of CBF verification for the returned tags, we only present the additional computational overhead for this part. It is important to note that although the number of sampled data blocks affects the final audit accuracy, with a damage rate of 1%, a sampling rate of 4.6% is already sufficient to maintain an audit probability of over 99%, which is considered adequate. In practical applications, e-commerce platforms can adjust the sampling rate according to their actual needs. In this study, the sampling rate is set at 4.6%. The additional time overhead is shown in Fig. 7.

As shown in Fig. 7, the time overhead required for CBF verification is very low, only in the millisecond range. Due to the combined effect of data block size and file size, the trend is similar to that observed in Fig. 5.

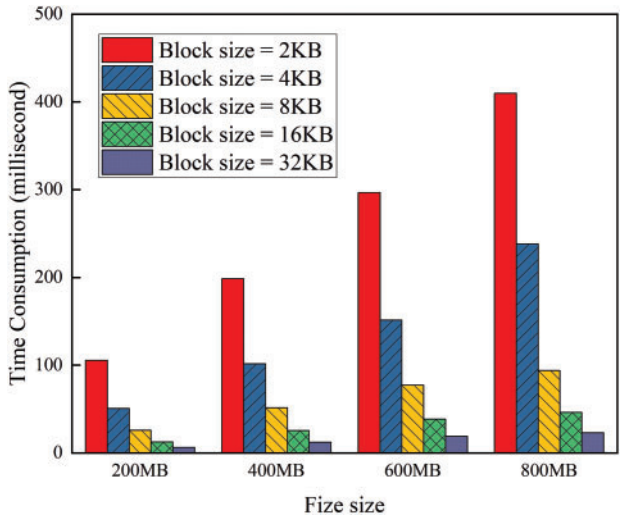

Figure 7: Extra computing overhead of Audit with different file size and block size

7 Discussion

This section discusses the extended usage features that OTGPDP can support and its existing limitations.

7.1 Extension of OTGPDP

First, OTGPDP can be extended to support batch auditing. Since the audit verification structure is $\prod_{i\in IND} H(filename||i)^{a_i} \times P = \sigma$, if there are multiple proofs $P = \{P^1, P^2, \ldots, P^m\}$, and corresponding signatures $\sigma = \{\sigma^1, \sigma^2, \ldots, \sigma^m\}$, the verification equation can be modified as follows: $\prod_{j=1}^{m} \prod_{i\in IND} H(filename||i)^{a_i} \times P^j = \prod_{j=1}^{m} \sigma^j$. This batch operation feature is better suited to multi-user scenarios, as it allows audit requests from multiple users to be packaged and verified together, thereby reducing communication and computational overhead.

Second, OTGPDP also supports the transition to public auditing. Public auditing allows entities other than the data owner to perform the auditing process. In OTGPDP, this can be achieved by using bilinear mappings and modifying the tag generation process. Specifically, during tag generation, set $t_i = H(filename||i) \times g^{d_i}$, and add the user's private key x as an exponent operation.

7.2 Limitation of OTGPDP

OTGPDP has a limitation: the size of the data blocks cannot exceed the set security parameter because the mathematical operations are performed in a prime cyclic group. To overcome this problem, RSA can be used as the basis for mathematical operations. This allows the data block size to be set arbitrarily. However, the downside of this approach is that the RSA parameter setting and subsequent computational overhead increases. Therefore, operational efficiency and the flexibility of an arbitrary data block size represent a trade-off that can be balanced based on the specific requirements of the application.

8 Conclusion and Future Work

Given the limited computational power of the e-commerce server, the proposed OTGPDP adopts the design approach of outsourcing audit metadata, using CBF for lightweight verification of the homomorphic verification tags generated by the cloud server. Additionally, by leveraging the updatable nature of CBF,

OTGPDP supports dynamic operations such as adding, deleting, and modifying data. Experiments demonstrate that OTGPDP does not introduce significant additional computational overhead to existing PDP protocols and alleviates the e-commerce server's burden of computing audit metadata. Therefore, OTGPDP is a valuable complement and optimization to existing PDP protocols.

In future work, we will further design PDP protocols that outsource audit metadata for multiple e-commerce servers.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by the Taiwan Comprehensive University System and the National Science and Technology Council of Taiwan under grant number NSTC 111-2410-H-019-006-MY3. Additionally, this work was financially/partially supported by the Advanced Institute of Manufacturing with High-tech Innovations (AIM-HI) from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan. Finally, the National Natural Science Foundation of China, No. 62402444; the Zhejiang Provincial Natural Science Foundation of China, No. LQ24F020012.

Author Contributions: The authors confirm their contribution to the paper as follows: data collection: Meiqin Tang; draft manuscript preparation: Yi Li, Wenying Zheng, and Yu-Sheng Su. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Ethics Approval: Not applicable.

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

Abbreviation List

PDP	Provable Data Possession
OTGPDP	PDP with Outsourced Tag Generation
CBF	Counting Bloom Filter
E-Commerce	Electronic-Commerce

References

1. Su YS, Hu YC. Applying cloud computing and Internet of Things technologies to develop a hydrological and subsidence monitoring platform. Sens Mater. 2022;34(4):1313. doi:10.18494/SAM3508.
2. Su YS, Hu YC, Wu YC, Lo CT. Evaluating the impact of pumping on groundwater level prediction in the Chuoshui River alluvial fan using artificial intelligence techniques. Int J Interact Multimed Artif Intell. 2024;8(7):28. doi:10.9781/ijimai.2024.04.002.
3. Chen B, Wu Z, Zhao R. From fiction to fact: the growing role of generative AI in business and finance. J Chin Econ Bus Stud. 2023;21(4):471–96. doi:10.1080/14765284.2023.2245279.
4. Jan Z, Ahamed F, Mayer W, Patel N, Grossmann G, Stumptner M, et al. Artificial intelligence for industry 4.0: systematic review of applications, challenges, and opportunities. Expert Syst Appl. 2023;216(9):119456. doi:10.1016/j.eswa.2022.119456.
5. Li Y, Shen J, Vijayakumar P, Lai CF, Sivaraman A, Sharma PK. Next-generation consumer electronics data auditing scheme toward cloud-edge distributed and resilient machine learning. IEEE Trans Consum Electron. 2024;70(1):2244–56. doi:10.1109/TCE.2024.3368206.
6. Bharadiya J. Artificial intelligence in transportation systems A critical review. Am J Comput Eng. 2023;6(1):34–45. doi:10.47672/ajce.1487.

7. Apell P, Eriksson H. Artificial intelligence (AI) healthcare technology innovations: the current state and challenges from a life science industry perspective. Technol Anal Strateg Manag. 2023;35(2):179–93. doi:10.1080/09537325.2021.1971188.
8. Necula SC, Păvăloaia VD. AI-driven recommendations: a systematic review of the state of the art in E-commerce. Appl Sci. 2023;13(9):5531. doi:10.3390/app13095531.
9. Raji MA, Olodo HB, Oke TT, Addy WA, Ofodile OC, Oyewole AT. E-commerce and consumer behavior: a review of AI-powered personalization and market trends. GSC Adv Res Rev. 2024;18(3):66–77. doi:10.30574/gscarr.2024.18.3.0090.
10. Whang SE, Roh Y, Song H, Lee JG. Data collection and quality challenges in deep learning: a data-centric AI perspective. VLDB J. 2023;32(4):791–813. doi:10.1007/s00778-022-00775-9.
11. Li Y, Shen J, Ji S, Lai YH. Blockchain-based data integrity verification scheme in AIoT cloud-edge computing environment. IEEE Trans Eng Manag. 2023;71(2014):12556–65. doi:10.1109/TEM.2023.3262678.
12. Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, et al. Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security; 2007 Oct 29–Nov 2; Alexandria, VA, USA. p. 598–609.
13. Erway CC, Küpçü A, Papamanthou C, Tamassia R. Dynamic provable data possession. ACM Trans Inf Syst Secur. 2015;17(4):1–29. doi:10.1145/2699909.
14. Li Y, Zheng W, Pandi V, Bhuiyan MZA, Thamilarasi C. Parallel and batch multiple replica auditing protocol for edge computing. In: 2023 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom); 2023 Dec 21–24; Wuhan, China. p. 254–61.
15. Bonomi F, Mitzenmacher M, Panigrahy R, Singh S, Varghese G. An improved construction for counting bloom filters. In: Algorithms-ESA 2006: 14th Annual European Symposium; 2006 Sep 11–13; Zurich, Switzerland. p. 684–95.
16. Verma A, Kuo YH, Kumar MM, Pratap S, Chen V. A data analytic-based logistics modelling framework for E-commerce enterprise. Enterp Inf Syst. 2023;17(6):2028195. doi:10.1080/17517575.2022.2028195.
17. Barata SFPG, Ferreira FAF, Carayannis EG, Ferreira JJM. Determinants of E-commerce, artificial intelligence, and agile methods in small- and medium-sized enterprises. IEEE Trans Eng Manag. 2023;71(1):6903–17. doi:10.1109/TEM.2023.3269601.
18. Ogbeyemi A, Odeyemi J, Igenewari O, Ogbeyemi A. A human factor approach to distribution network design for e-commerce in supply chain system: a case study. Enterp Inf Syst. 2023;17(12):2200767. doi:10.1080/17517575.2023.2200767.
19. Yu HY, Ogbeyemi A, Lin WJ, He J, Sun W, Zhang WJ. A semantic model for enterprise application integration in the era of data explosion and globalisation. Enterp Inf Syst. 2023;17(4):1989495. doi:10.1080/17517575.2021.1989495.
20. Wang C, Ahmad SF, Bani Ahmad Ayassrah AYA, Awwad EM, Irshad M, Ali YA, et al. An empirical evaluation of technology acceptance model for Artificial Intelligence in E-commerce. Heliyon. 2023;9(8):e18349. doi:10.1016/j.heliyon.2023.e18349.
21. Upadhyay U, Kumar A, Sharma G, Sharma S, Arya V, Panigrahi PK, et al. A systematic data-driven approach for targeted marketing in enterprise information system. Enterp Inf Syst. 2024;18(8):2356770. doi:10.1080/17517575.2024.2356770.
22. Tsang YP, Fan Y, Lee CKM, Lau HCW. Blockchain sharding for e-commerce supply chain performance analytics towards Industry 5.0. Enterp Inf Syst. 2024;18(4):2311807. doi:10.1080/17517575.2024.2311807.
23. Shen J, Zeng P, Choo KR, Li C. A certificateless provable data possession scheme for cloud-based EHRs. IEEE Trans Inf Forensics Secur. 2023;18:1156–68. doi:10.1109/TIFS.2023.3236451.
24. Yang Y, Chen Y, Chen F, Chen J. An efficient identity-based provable data possession protocol with compressed cloud storage. IEEE Trans Inf Forensics Secur. 2022;17:1359–71. doi:10.1109/TIFS.2022.3159152.
25. Wang L, Hu M, Jia Z, Guan Z, Chen Z. SStore: an efficient and secure provable data auditing platform for cloud. IEEE Trans Inf Forensics Secur. 2024;19:4572–84. doi:10.1109/TIFS.2024.3383772.

26. Li T, Wang H, He D, Yu J. Synchronized provable data possession based on blockchain for digital twin. IEEE Trans Inf Forensics Secur. 2022;17:472–85. doi:10.1109/TIFS.2022.3144869.
27. Deng L, Wang B, Wang T, Feng S, Li S. Certificateless provable data possession scheme with provable security in the standard model suitable for cloud storage. IEEE Trans Serv Comput. 2023;16(6):3986–98. doi:10.1109/TSC.2023.3303185.
28. Wang H, Wan Z, He D, Yu J. Synchronous blockchain-based distributed provable data possession with forward-security. IEEE Trans Serv Comput. 2024;17(3):1227–38. doi:10.1109/TSC.2023.3324023.
29. Dhakad N, Kar J. EPPDP: an efficient privacy-preserving data possession with provable security in cloud storage. IEEE Syst J. 2022;16(4):6658–68. doi:10.1109/JSYST.2022.3159847.
30. Luo L, Guo D, Ma RTB, Rottenstreich O, Luo X. Optimizing bloom filter: challenges, solutions, and comparisons. IEEE Commun Surv Tutor. 2018;21(2):1912–49. doi:10.1109/COMST.2018.2889329.
31. Tarkoma S, Rothenberg CE, Lagerspetz E. Theory and practice of bloom filters for distributed systems. IEEE Commun Surv Tutor. 2012;14(1):131–55. doi:10.1109/SURV.2011.031611.00024.
32. Pelle I, Czentye J, Doka J, Sonkoly B. Towards latency sensitive cloud native applications: a performance study on AWS. In: 2019 IEEE 12th International Conference on Cloud Computing (CLOUD); 2019 Jul 8–13; Milan, Italy. p. 272–80.