Hindawi
Security and Communication Networks
Volume 2021, Article ID 6639634, 11 pages
https://doi.org/10.1155/2021/6639634

Research Article

Efficient Privacy-Preserving Certificateless Public Auditing of Data in Cloud Storage

Hao Yan,[1,2] **Yanan Liu**,[1] **Zheng Zhang**,[1] **and Qian Wang**[3]

[1]*School of Network Security, Jinling Institute of Technology, Nanjing 211169, Jiangsu, China*
[2]*Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350007, Fujian, China*
[3]*The 28th Research Institute of China Electronics Technology Group Corporation, Nanjing 210007, Jiangsu, China*

Correspondence should be addressed to Hao Yan; pxy_hao@163.com

Received 23 December 2020; Revised 4 April 2021; Accepted 13 May 2021; Published 28 May 2021

Academic Editor: Fulvio Valenza

Cloud computing is a fast-growing technology which supplies scalable, innovative, and efficient business models. However, cloud computing is not fully trusted, and the security of the data outsourced in cloud storage needs to be guaranteed. One of the hottest issues is how to ensure the integrity of the data in cloud storage. Until now, many researchers have proposed lots of provable data possession (PDP) schemes to deal with the problem of data integrity audition. Nevertheless, very little effort has been devoted to preserve the data uploader's privacy while auditing the integrity of data shared in a group. To overcome the shortcoming, we propose a novel certificateless PDP protocol to efficiently audit the integrity of data shared in a workgroup with user privacy preserving. Due to the inherent structural advantage of the certificateless crypto mechanism, our PDP scheme eliminates the key escrow problem and the certificate management problem simultaneously. Moreover, the audition process in our scheme does not need any user's identity which helps to keep the anonymity of data uploader. We give for our scheme a detailed security proof and efficiency analysis. Experiment results of performance evaluation demonstrate that our new scheme is very efficient and feasible.

1. Introduction

Recently, cloud computing has continued to provide scalable and low-cost services to user. The core advantage of cloud storage is dynamic scalability that allows the cloud storage services to deal with increasing amounts of data. Therefore, a vast number of organizations and people would like to buy cloud storage service for data maintenance and management as one of fundamental investments. Moreover, with cloud storage platform, users are easy to work together in one team [1–4], in which they share data with each other. However, cloud service provider (CSP) is not fully trustworthy. The data stored in CSP might be corrupted or deleted because of accidental hardware errors, network exceptions, software bugs, or human mistakes [5–8]. Furthermore, the untrusted CSP can tamper the user's data easily by either deleting or modifying them. To escape economic compensation and keep good reputation, CSP would not tell the truth to user. Additionally, with no audition mechanism, untrusted CSP can never be detected. Therefore, cloud users need to periodically audit whether the data outsourced in cloud storage server is kept well.

The PDP model supplies the user an efficient method to audit the integrity of the remote data in cloud storage. The audition process of PDP is conducted by a challenge-response mechanism. In PDP schemes, the data owner divides their data to many small data blocks and binds one tag to each data block. Since the tag contains the information of data block, user can get the integrity status of data block through checking the validity of the corresponding tag. Until now, many articles have proposed several types of PDP schemes [9–39] for different application scenarios. However, most PDP protocols are just suitable for checking the integrity of single data that belong to only one user.

In real applications, sharing data among multiple users is a common situation, in which the shared data can be used by any one of the workgroup. Therefore, auditing the integrity of data shared in a workgroup is an essential task which should be

solved by PDP scheme. When auditing shared data, user anonymity against third party auditor (TPA) is an important security requirement. In practice, TPA is usually assumed to be honest-but-curious, which means TPA tries to guess the identity of data uploader when auditing the data integrity. If the identity is exposed, the data uploader may face great security threats especially when the data are sensitive. For example, every person can report to the government about criminal behaviors through open complaint platform. If the criminal knows who reported his behavior, he may revenge the reporter. To prevent criminal from revenging the reporter, it is necessary to preserve reporter's identity privacy. Therefore, PDP scheme should keep confidential of uploader's identity to TPA. Aim to this goal, Wang et al. [23] proposed a concrete PDP protocol with the notion of user privacy preserving for shared data. Following, several schemes [24–29] with user privacy preserving are proposed. However, most previous PDP schemes are constructed by the PKI technique which suffers from certificate management problems such as generation, distribution, renew, revocation, update, and verification. To avoid certificate management, some PDP schemes are designed based on identity-based public cryptography (IBC) [40]. However, IBC also has the natural drawback of "key escrow." To address these shortcomings, certificateless cryptography (CLC) [41] is introduced as a cryptography primitive. In CLC, user's private key is consisted of two components: the first is the partial key and the second is the secret value. User's partial key is computed by the key generation center (KGC), but the secret value is computed by the user himself/herself which is unknown to KGC. Therefore, CLC overcomes the drawbacks of PKI and IBC simultaneously. Because of these advantages, some researchers utilize CLC to construct PDP schemes [31–39]. Nevertheless, these schemes also have other shortcomings such as no user privacy preserving, heavy computationally cost, or existing security flaws which reduce the practicability of the schemes. Thus, it is necessary and urgent to present more efficient and secure PDP scheme based on CLC with user privacy preserving.

1.1. Our Contributions. Most previous PDP schemes only concentrate on verifying the integrity of personal data. However, to share data with multiple users based on cloud platform is a development trend and is becoming popular. Because any user can upload data to the cloud, the privacy of data uploader's identity should be guaranteed. That is to say, TPA can audit data integrity with the help of CSP but cannot distinguish the exact data uploader.

In this manuscript, we mainly consider to verify the integrity of data shared in a group with user privacy preserving. Our primary contributions in this study are summarized as following.

(1) We present the security model of certificateless-based PDP scheme for group shared data with user privacy protection. It defines the abilities of adversaries and the requirement of user privacy preserving.

(2) We propose the concrete PDP scheme based on CLC for group shared data with user privacy preserving. The proposal can resist the attacks of two types of adversaries and keep user privacy against TPA.

(3) We give rigorous secure proofs to prove the security of the proposed scheme in a random oracle model. We also demonstrate the performance evaluation results of our scheme and make comprehensive comparisons with several existing schemes.

1.2. Related Work. The initial PDP model is proposed by Ateniese et al. [9], which tried to provide a method to verify the integrity of client's data stored in a remote server without downloading the data. To get better efficiency, they realized blockless verification by using homomorphic verifiable tags. Furthermore, they proposed two concrete schemes based on the RSA algorithm. However, the schemes were only available for static data with no support for dynamic operations. With the aim to enhance the scalability, Ateniese et al. [10] extended their initial PDP schemes and proposed an improved one based on symmetric key encryption. Although the improved scheme realized dynamic data operations as appending, updating, and deleting, the drawbacks still existed that the challenge number of the scheme was limited and did not support data inserting. Subsequently, Juels and Kaliski [11] proposed a similar model called proof of retrievability (POR) which had error-correcting capabilities besides data integrity audition. To improve efficiency, Shacham and Waters [12] developed a compact PoR scheme with a shorter authentication tag.

Later, Erway et al. [13] presented a PDP scheme which supported public integrity audition and fully data dynamic operations. To improve the efficiency of dynamic operation, Yan et al. [14] realized a PDP scheme with a new data structure that stored all blocks operation records. To increase data durability, Liu et al. [15] presented a multireplicas data integrity checking scheme, which supported fully dynamic data updates. Li et al. [16] further considered a more complex environment that multicopies were stored in multi-CSPs, and they constructed a concrete scheme to check the integrity of all copies for one time. In other works, Wang [17] proposed a proxy PDP scheme in which a commitment was used to authenticate the validity of the auditor. Yan et al. [18] strengthened the restriction for the verifier and proposed a verifier-designated PDP protocol. Wang et al. [19] presented a notion of data privacy protection and designed a public auditable PDP scheme. Shen et al. [20] designed a PDP protocol to guarantee the privacy of authenticators.

In recent years, many cloud applications supported users to work in coordination with shared data. Therefore, how to audit the data shared among multiusers attracted many attentions. Wang et al. [21] designed the first PDP scheme by a ring signature technique to verify the integrity of data shared in a group with multiusers. The scheme also supported public auditing and user privacy preserving. Later, Yang et al. [22] proposed a PDP protocol for group data with user identity privacy and traceability. Wang et al. [23] designed a new PDP scheme to support dynamic groups which allowed group members to join or leave the group at

any time. Wu et al. [24] developed a PDP scheme for auditing the integrity of data shared within multiple uploaders. Subsequently, Wang et al. [25] presented a PDP protocol based on the proxy resignature technique to address the problem of user revocation. Nonetheless, all these PDP schemes were designed by the traditional PKI mechanism which bears heavy cost of certificate management.

To eliminate certificate management, the identity-based cryptography (IBC) mechanism is used by many researchers to construct PDP schemes. Until now, several IBC-based PDP schemes have been proposed. For instance, Wang et al. [26] designed the first IBC-based data integrity checking scheme and proved its security under the defined security model. Yu et al. [27] presented an IBC-based PDP scheme which supported the dynamic group and data privacy protection. Tan and Jia [28] relied on an IBC-based signature scheme to propose a PDP scheme which also alleviated the users' fear of losing their keys. To improve the applicability of cloud storage, Zhang et al. [29] proposed a proxy-oriented identity-based encryption with a keyword search scheme from lattices for cloud storage, which was postquantum secure. Furthermore, Zhang et al. [29] proposed a scheme CIPPPA to check the integrity of medical data generated by wireless body area networks (WBANs). CIPPPA can not only achieve conditional identity privacy of patients in WBANs but also validate malicious auditing behaviors with the help of ethereum blockchain.

Unfortunately, IBC also has its own inherent drawback named "key escrow." To address this problem, PDP schemes based on CLC were proposed in many articles. Wang et al. [31] first presented a CLC-based PDP scheme for auditing cloud data. In this scheme, KGC computed the partial key for each user, but KGC did not know the user's secret value, so the user's private key was protected against KGC which avoided the key escrow problem. However, He et al. [32] thought the scheme in [31] is insecure because it did not give the formal security model. Subsequently, they proposed a CLC-based PDP scheme for checking the data of WBANs. Nevertheless, this scheme is proved insecure [33] either. To improve verification efficiency, Kim and Jeong [34] proposed a CLC-based PDP scheme with constant verification time. Similarly, Yang et al. [35] presented a PDP scheme for shared data integrity audition based on certificateless cryptography. The scheme claimed that it was able to guarantee user identity, but in the verification phase, TPA got the relationship between data and the public keys. Thus, it did not really realize user privacy preserving. Li et al. [36] presented a PDP protocol of group shared data based on certificateless cryptography, but the scheme lost the user privacy preservation feature. Kang et al. [37] proposed a certificateless public auditing scheme with privacy preserving for cloud-assisted WBANs which protected the data from being directly exposed to the TPA. Ming and Shi [38] proposed an efficient CLC-based PDP scheme with user privacy protection. Wu et al. [39] also designed a PDP scheme for multiusers setting with user privacy preserving, but the overheads of both communication and computation were too heavy especially in the challenge phase.

2. Preliminaries

We first review some preliminary cryptography knowledge throughout this study.

2.1. Bilinear Maps. Assume that two multiplicative cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ have large prime order q. Let $g \in \mathbb{G}_1$ to be one generator of $\mathbb{G}_1$. Define e: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ is a bilinear map with the following properties.

(a) Computability: for any $u, v \in \mathbb{G}_1$, and there exist efficient algorithms to calculate the value of $e(u, v)$.

(b) Bilinearity: for any $x, y \in Z_q^*$ and $u, v \in \mathbb{G}_1$, and it has $e(u^x, v^y) = e(u, v)^{xy}$.

(c) Nondegeneracy: $\exists u, v \in \mathbb{G}_1$, so that $e(u, v) \neq 1_{\mathbb{G}_2}$.

2.2. Assumption

Definition 1. Computational Diffie–Hellman assumption: g is a generator of the multiplicative cyclic group $\mathbb{G}_1$. Given (g, g^a, g^b), to get g^{ab} is computationally intractable with unknown $a, b \in Z_q^*$. For any adversary $\mathscr{A}$, the probability for $\mathscr{A}$ to solve this problem is negligible. We define the CDH problem as

$$P_r\left[\mathscr{A}^{\mathscr{CDH}}\left(g, g^a, g^b\right) = g^{ab} \in \mathbb{G}_1 : a, b \overset{R}{\leftarrow} Z_q\right] \leq \varepsilon. \quad (1)$$

3. System Model and Security Model

3.1. System Model. There are four participants in our scheme: KGC, CSP, user group, and TPA.

(1) KGC is a trusted organization which generates the partial key for user. We assume the partial key is transmitted by secure channels.

(2) CSP is the cloud storage service provider who maintains user's data and generates integrity proofs to prove the data integrity when received the challenge from TPA.

(3) A user group has several users, and every user can upload data blocks to CSP by which all users share their data to each other.

(4) TPA is responsible for auditing the integrity of data shared in a group. TPA sends an integrity challenge to CSP and gets a proof from CSP. Then, TPA validates the rightness of the proof and informs the checking result to users.

The system model of the proposed scheme is shown in Figure 1. It assumed that CSP is semitrusted. Namely, it can execute audition protocol honestly, but lies to TPA when data are broken. TPA is honest-but-curious, that is, TPA audits the data integrity honestly and responds the real audition result to data user, but it is curious about the identity of data uploader.

Our certificateless auditing scheme for group shared data with user privacy preserving consisted of seven algorithms:

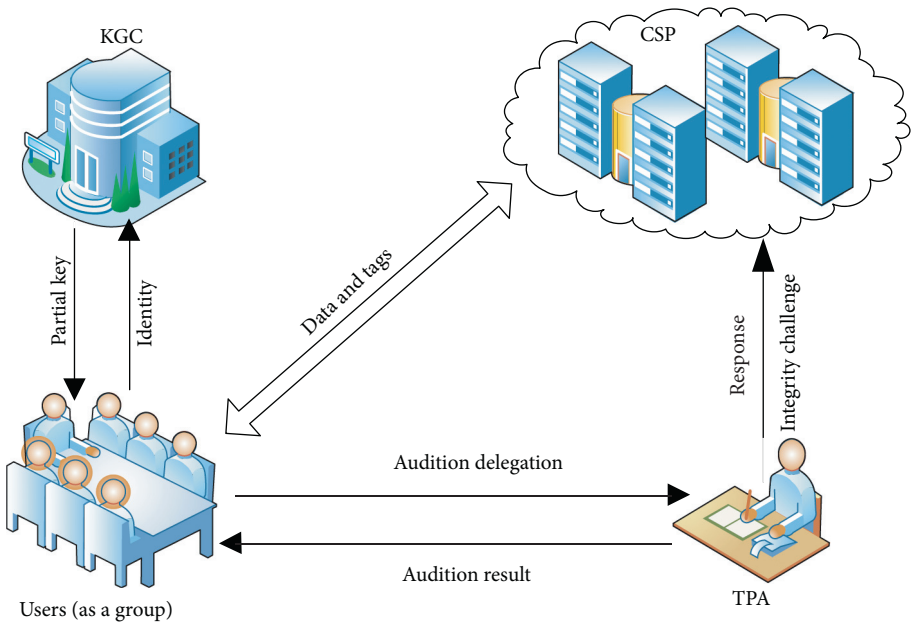

Figure 1: System model of our scheme.

Setup, PartialKeyGen, SecretValueGen, PublicKeyGen, TagGen, Challenge, Proof, and Audit.

Setup: with the security parameter k, this algorithm generates public parameters pp and master private key msk.

PartialKeyGen: KGC runs this algorithm to compute user's partial key. It inputs the identity id_i of the user u_i and outputs u_i's partial key d_i.

SecretValueGen: each user (u_i) performs this algorithm to compute the secret value (s_i)

PublicKeyGen: each user (u_i) performs this algorithm to compute the public key (PID_i)

TagGen: this algorithm generates an authentication tag for each data block. It inputs user u_i's secret key $sk_i = (d_i, s_i)$, and the block m_j outputs its tag $T_{i,j}$.

Challenge: this algorithm is performed by TPA to select a data integrity challenge chal

Proof: the algorithm generates the data integrity proof P for each challenge chal. It takes the inputs of shared data F, tags collection T, and the challenge chal.

Audit: this algorithm is used to audit the rightness of integrity proof. It takes the inputs of the challenge chal, proof P, and data identity Fid. If P passes the verification, the algorithm returns "1;" otherwise, it returns "0."

3.2. Security Model. Referring to [32, 42], the security model of our proposed scheme contains two types of adversaries. The first one denoted by $\mathscr{A}_1$ cannot access the master key but can replace the user's public key. The second one denoted by $\mathscr{A}_2$ knows the master key but cannot replace the user's public key. We utilize a game to cover the security characters of our scheme; the game involves a super adversary $\mathscr{A} \in \{\mathscr{A}_1, \mathscr{A}_2\}$ and a challenger $\mathscr{C}$.

Setup phase: $\mathscr{C}$ calls Setup to generate the master private key msk and the public parameter pp. If $\mathscr{A}$ is the first type adversary $\mathscr{A}_1$, $\mathscr{C}$ gives pp to $\mathscr{A}$. If $\mathscr{A}$ is the second type adversary $\mathscr{A}_2$, $\mathscr{C}$ gives both the pp and msk to $\mathscr{A}$.

Queries phase: $\mathscr{A}$ makes four types of query to $\mathscr{C}$ for polynomial times. $\mathscr{C}$ returns the results to $\mathscr{A}$.

(a) Hash query. Adversary $\mathscr{A}$ queries about hash values of any hash function in the scheme. $\mathscr{C}$ replies the hash value to $\mathscr{A}$.

(b) Partial key query. Adversary $\mathscr{A}$ can query any user's partial key with the identity id_i. $\mathscr{C}$ calculates the partial key d_i by the algorithm PartialKeyGen and returns d_i to $\mathscr{A}$ (this step is executed only by the first type adversary $\mathscr{A}_1$).

(c) Secret value query. $\mathscr{A}$ can query any user's secret value with any identity id_i. $\mathscr{C}$ computes the secret value s_i by the algorithm SecretValueGen and returns s_i to $\mathscr{A}$.

(d) Tag query. Adversary $\mathscr{A}$ can randomly select blocks and query their tags generated by any user in the group. $\mathscr{C}$ generates the tag of the queried block and sends the tag back to $\mathscr{A}$. If $\mathscr{C}$ does not have user's private key in this step, he can compute the key by PartialKeyGen and SecretValueGen algorithms.

Public key replacement: $\mathscr{C}$ can change any user's public key to any other value (this step is executed only by $\mathscr{A}_1$)

Forge phase: finally, $\mathscr{A}$ submits to $\mathscr{C}$ a forged proof P^* for any (id_i^*, m_j^*) with the public key PID_i^*. If the proof satisfies the following three conditions, $\mathscr{A}$ wins this game.

(1) P^* passes the audition with id_i^* and PID_i^*
(2) If $\mathscr{A}$ is the first type adversary $\mathscr{A}_1$, the partial key and secret value of id_i^* have not been queried. If $\mathscr{A}$ is the second type adversary $\mathscr{A}_2$, the secret value of id_i^* has never been queried.
(3) m_j^* has never been performed the tag query with user identity id_i^* and PID_i^*

Definition 2. A public certificateless PDP scheme for group shared data with user privacy preserving is secure, if for any adversary $\mathscr{A}$, to win the game above only with negligible probability.

User privacy preserving is another security feature of the scheme. Since multiple users share data with each other in a group, each one can upload data to the group. In many cases, users prefer to keep anonymous against TPA. An honest-but-curious TPA tries to distinguish the identity of data uploader during the data verification process. If the user information is revealed and leaked by TPA, the data uploader may face potential security threats. Thus, the scheme should guarantee user's anonymity against TPA.

Definition 3. A public certificateless PDP scheme for group shared data is user privacy preserving, if no information about the user identity is revealed by TPA within the procedure of data audition.

4. Construction of Our Scheme

We show the detailed construction of our certificateless PDP scheme for group shared data, which realizes public verification and user privacy protection.

Suppose the data F is shared in a group with N users denoted as $\{u_1, u_2, \ldots, u_N\}$. F is split into n blocks, and each block is denoted by m_i, where i is the block index. Different blocks may be uploaded by different users. The algorithms in our scheme are defined as follows.

$\mathrm{Setup}(1^k) \longrightarrow (\mathrm{pp}, \mathrm{msk})$: KGC first sets the value of security parameter k and selects a big random prime number q with $|q| = k$. Select cyclic multiplicative groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with order q and a bilinear map e: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$. KGC selects a generator g of $\mathbb{G}_1$ and three different hash functions: h_1: $\{0,1\} \longrightarrow \mathbb{G}_1$, h_2: $\{0,1\} \longrightarrow \mathbb{G}_1$, and h_3: $\{0,1\} \longrightarrow \mathbb{G}_1$. Then, KGC randomly selects $x \in Z_q^*$ and sets the master private key $\mathrm{msk} = x$, so the master public key is $P_0 = g^x$. The system parameter is $\mathrm{pp} = (q, g, \mathbb{G}_1, \mathbb{G}_2, e, P_0, h_1, h_2, h_3)$.

$\mathrm{PartialKeyGen}(\mathrm{id}_i) \longrightarrow d_i$: on receiving the identity id_i of the u_i, KGC computes $d_i = h_1(\mathrm{id}_i)^x$ as u_i's partial key and sends it to u_i by a secure channel

$\mathrm{SecretValueGen} \longrightarrow s_i$: u_i randomly selects a value $\lambda_i \in Z_q^*$ and sets the secret value $s_i = \lambda_i$

$\mathrm{PublicKeyGen} \longrightarrow \mathrm{PID}_i$: with secret value s_i, u_i computes the public key $\mathrm{PID}_i = g^{1/s_i}$

$\mathrm{TagGen}(d_i, s_i, m_j) \longrightarrow T_{i,j}$: u_i computes the value $U = h_3(P_0, n)$ and generates the tag for the block m_j by the following equation.

$$T_{i,j} = \left(d_i \cdot h_2(\mathrm{Fid}, j) \cdot U^{m_j}\right)^{s_i}. \tag{2}$$

Here, Fid is the unique identification of the data F. Finally, u_i uploads $(m_j, T_{i,j}, \mathrm{id}_i)$ to CSP. CSP validates the rightness of the tag by the following equation:

$$e\left(T_{i,j}, \mathrm{PID}_i\right) = e\left(h_1(\mathrm{id}_i), P_0\right) \cdot e\left(h_2(\mathrm{Fid}, j) \cdot U^{m_j}, g\right). \tag{3}$$

It can be confirmed as follows:

$$\begin{aligned} e\left(T_{i,j}, \mathrm{PID}_i\right) &= e\left(\left(h_1(\mathrm{id}_i)^x \cdot h_2(\mathrm{Fid}, j) \cdot U^{m_j}\right)^{\lambda_i}, g^{1/\lambda_i}\right) \\ &= e\left(h_1(\mathrm{id}_i)^x, g\right) \cdot e\left(h_2(\mathrm{Fid}, j) \cdot U^{m_j}, g\right) \\ &= e\left(h_1(\mathrm{id}_i), P_0\right) \cdot e\left(h_2(\mathrm{Fid}, j) \cdot U^{m_j}, g\right). \end{aligned} \tag{4}$$

$\mathrm{Challenge}(\mathrm{Fid}) \longrightarrow \mathrm{chal}$: to challenge the integrity of the data named Fid, TPA randomly chooses c numbers from the set $\{1, 2, \ldots, n\}$ to get a subset $C \subseteq \{1, 2, \ldots, n\}$, where $|C| = c$. For each number $l \in C$, CSP randomly selects a value $v_l \in Z_q^*$ and sets the $\mathrm{chal} = \{(l, v_l) | l \in C\}$. TPA submits $\mathrm{chal} = \{(l, v_l) | l \in C\}$ to CSP.

$\mathrm{Proof}(F, T, \mathrm{chal}) \longrightarrow P$: with $\mathrm{chal} = \{(l, v_l) | l \in C\}$, CSP finds out all the challenged tuples $\Theta = \{(m_l, T_{i,l}, u_i) | l \in C\}$. Then, CSP randomly selects a value $\alpha \in \mathbb{G}_1$ and computes

$$\begin{aligned} \sigma_1 &= \alpha \cdot \prod_{(m_l, T_{i,l}, \mathrm{id}_i) \in \Theta} h_1(\mathrm{id}_i)^{v_l}, \\ \sigma_2 &= e(\alpha, P_0) \cdot \prod_{(m_l, T_{i,l}, \mathrm{id}_i) \in \Theta} e\left(T_{i,l}^{v_l}, \mathrm{PID}_i\right), \\ M &= \sum_{(m_l, T_{i,l}, \mathrm{id}_i) \in \Theta} v_l m_l. \end{aligned} \tag{5}$$

Finally, CSP sends the proof $P = (\sigma_1, \sigma_2, M)$ to TPA

$\mathrm{Audit}(\mathrm{chal}, P, \mathrm{Fid}) \longrightarrow \{0,1\}$: when receiving P returned from CSP, TPA computes $U = h_3(P_0, n)$ and checks the following equation:

$$\sigma_2 = e(\sigma_1, P_0) \cdot e\left(\prod_{(m_l, T_{i,l}, \mathrm{id}_i) \in \Theta} h_2(\mathrm{Fid}, l)^{v_l} \cdot U^M, g\right). \tag{6}$$

If equation (6) holds, returns 1; otherwise, returns 0.

The equation (6) can be confirmed as follows:

$$\begin{aligned}
\sigma_2 &= e(\sigma_1, P_0)\cdot e\left(\prod_{(m_l,T_{i,l},\mathrm{id}_i)\in\Theta} h_2(\mathrm{Fid},l)^{v_l}\cdot U^M, g\right)\\
&= e(\alpha, P_0)\cdot e\left(\prod_{(m_l,T_{i,l},\mathrm{id}_i)\in\Theta} h_1(\mathrm{id}_i)^{v_l}, g^x\right)\cdot e\left(\prod_{(m_l,T_{i,l},\mathrm{id}_i)\in\Theta} h_2(\mathrm{Fid},l)^{v_l}\cdot U^{\sum_{(m_l,T_{i,l},\mathrm{id}_i)\in\Theta} v_l m_l}, g\right)\\
&= e(\alpha, P_0)\cdot e\left(\prod_{(m_l,T_{i,l},\mathrm{id}_i)\in\Theta} h_1(\mathrm{id}_i)^{xv_l}\cdot \prod_{(m_l,T_{i,l},\mathrm{id}_i)\in\Theta} (h_2(\mathrm{Fid},l)\cdot U^{m_l})^{v_l}, g\right)\\
&= e(\alpha, P_0)\cdot e\left(\prod_{(m_l,T_{i,l},\mathrm{id}_i)\in\Theta} (h_1(\mathrm{id}_i)^{x}\cdot h_2(\mathrm{Fid},l)\cdot U^{m_l})^{v_l}, g\right)\\
&= e(\alpha, P_0)\cdot e\left(\prod_{(m_l,T_{i,l},\mathrm{id}_i)\in\Theta} (h_1(\mathrm{id}_i)^{x}\cdot h_2(\mathrm{Fid},l)\cdot U^{m_l})^{\lambda_i v_l}, g^{1/\lambda_i}\right)\\
&= e(\alpha, P_0)\cdot \prod_{(m_l,T_{i,l},\mathrm{id}_i)\in\Theta} e\left(T_{i,j}, \mathrm{PID}_i\right).
\end{aligned} \tag{7}$$

5. Security Proof

In this section, we show the security proof of our new scheme under the security model defined in Section 3.2. In our proof, three hash functions (h_1, h_2, h_3) used in our scheme are all random oracles.

Lemma 1. *If the CDH problem $(g, \mathbb{G}_1, g^a, g^b)$ is hard for the group $\mathbb{G}_1$, our scheme is secure against $\mathscr{A}_1$.*

Proof. If the adversary $\mathscr{A}_1$ wins the game, a simulator β can be designed to solve the CDH hard problem resorting to $\mathscr{A}_1$. Let $(g, \mathbb{G}_1, g^a, g^b)$ to be one CDH instance, and β computes g^{ab} by following steps.

Setup: β sets the master public key $P_0 = g^a$, where a is unknown to β. β randomly selects public parameters and gives them to $\mathscr{A}_1$.

h_1 query: $\mathscr{A}_1$ adaptively queries the hash value of any identity id^*. β keeps a table $\mathrm{tab}_1 = \{(\mathrm{id}, \overline{h_1}, q_1, \tau)\}$ for the h_1 query. If tab_1 contains the row $(\mathrm{id}, *, *, *)$, β gets the row $(\mathrm{id}^*, \overline{h_1^*}, q_1^*, \tau^*)$ from tab_1 and returns q_1^* to $\mathscr{A}_1$. Otherwise, β selects a random number $\overline{h_1^*} \in Z_q^*$ and tosses a coin $\tau \in \{0, 1\}$. Suppose the probability of $\tau = 1$ is γ and the probability of $\tau = 0$ is $1 - \gamma$. If $\tau = 1$, β computes $q_1^* = (g^b)^{\overline{h_1^*}}$. Otherwise, β computes $q_1^* = g^{\overline{h_1^*}}$. β responds q_1^* to $\mathscr{A}_1$ and appends a new row $(\mathrm{id}^*, \overline{h_1^*}, q_1^*, \tau^*)$ in table tab_1.

Partial key query: $\mathscr{A}_1$ sends any identity id^* to β for querying the partial key. β maintains a table $\mathrm{tab}_2 = \{(\mathrm{id}, d, s, pk)\}$ and searches the row $(\mathrm{id}^*, d^*, *, *)$. If the row exists, β returns d^* to $\mathscr{A}_1$. If the row does not exist, β first gets $(\mathrm{id}^*, \overline{h_1^*}, q_1^*, \tau^*)$ from tab_1. If $\tau^* = 1$, β aborts and exits the game, and if $\tau^* = 0$, β sets $d^* = (q_1^*)^a = (g^a)^{\overline{h_1^*}}$ and inserts the new row $(\mathrm{id}^*, d^*, \perp, \perp)$ to tab_2.

Secret value query: with the query id^*, β searches the row $(\mathrm{id}^*, *, s^*, *)$. If the row exists, β sends s^* to $\mathscr{A}_1$. Otherwise, $\mathscr{B}$ randomly chooses a value $s^* \in Z_q^*$, returns s^* to $\mathscr{A}_1$, and inserts the row $(\mathrm{id}^*, \perp, s^*, g^{1/s^*})$ to tab_2.

Public key query: for the queried id^*, β searches the row $(\mathrm{id}^*, *, s^*, pk^*)$; if it exists, β returns pk^* to $\mathscr{A}_1$. Otherwise, $\mathscr{B}$ selects a random value $s^* \in Z_q^*$, sends $g^{1/s*}$ to $\mathscr{A}_1$, and inserts the row $(\mathrm{id}^*, \perp, s^*, g^{1/s*})$ to tab_2.

Public key replacement: $\mathscr{A}_1$ sends $(\mathrm{id}^*, \mathrm{ns}^*, \mathrm{npk}^*)$ to β to replace user's (id^*) public key with new value $(\mathrm{ns}^*, \mathrm{npk}^*)$. β searches the row $(\mathrm{id}^*, *, s^*, \mathrm{pk}^*)$; if it exists, β returns and update the row to $(\mathrm{id}^*, *, \mathrm{ns}^*, \mathrm{npk}^*)$. If the row does not exist, β insert a new row $(\mathrm{id}^*, \perp, \mathrm{ns}^*, \mathrm{npk}^*)$ to tab_2.

h_2 query: $\mathscr{A}_1$ can query the hash value of (Fid, j) at any time. For this query, β keeps a list tab_3 with tuple (Fid, j, q_2). If the row $(\mathrm{Fid}, j, *)$ exists in tab_3, β retrieves q_2 and returns it to $\mathscr{A}_1$. Otherwise, β randomly chooses a value $q_2^* \in \mathbb{G}_1$ and returns $q_2{}^*$ to $\mathscr{A}_1$. β inserts a new row (Fid, j, q_2^*) into tab_3.

h_3 query: $\mathscr{A}_1$ can query the hash value of (P_0, v) at any time. For this query, β keeps a list tab_4 with tuple (P_0, v, q_3). If the row $(P_0, v, *)$ exists in tab_4, β

retrieves q_3 and returns it to $\mathscr{A}_1$. Otherwise, β randomly chooses $q_3^* \in \mathbb{G}_1$ and returns q_3^* to $\mathscr{A}_1$. Then, β inserts a new row (P_0, v, q_3^*) into tab_4.

Tag query: for the tag query $(\text{Fid}, j, m_j, \text{id}, v)$, β gets the row $(\text{id}, \overline{h_1}, q_1, \tau)$ from tab_1. If $\tau = 1$, β aborts and exits. Otherwise, β searches (id, d, s, pk), (Fid, j, q_2), and (P_0, v, q_3) from tab_2, tab_3, and tab_4, respectively. β computes the tag: $T_{i,j} = (d \cdot q_2 \cdot q_3^{m_j})^s = (g^{a\overline{h_1}} \cdot q_2 \cdot q_3 m_j)^s$ and returns it to $\mathscr{A}_1$.

Forge: at last, $\mathscr{A}_1$ gives a forged tag $T'_{i*,j*}$ for block m'_{j*} with the identity id_{i*}', block number v', and npk'_{i*}. It is restricted that m'_{j*} has not executed the tag query under such conditions before.

Analysis: it is not difficult to see that if $\mathscr{A}_1$ wins the game, the equation $e(T'_{i*,j*}, \text{PID}_{i*}) = e(h_1(\text{id}_{i*}'), g^a) \cdot e(h_2(\text{Fid}, j*) \cdot U^{m_{j*}'}, g)$ must hold according to equation (3). Then, β gets the row $(\text{id}_{i*}', \overline{h_1}', q_1', \tau')$ from tab_1. If $\tau' = 0$, β aborts and exits. Otherwise, β continues to find the row $(\text{id}_{i^{*\prime}}, d', \text{ns}', \text{npk}')$ from tab_2, $(\text{Fid}, j*, q_2')$ from tab_3, and (P_0, v', q_3') from tab_4. Thus, the equation above can be changed to $e(T'_{i*,j*}, g^{1/ns'}) = e(g^{ab\overline{h_1}'} \cdot q_2' \cdot q_3^{'m_{j*}'}, g)$. We can compute the result of given CDH instance: $g^{ab} = ((T'_{i*,j*})^{1/ns'} / (q_2' \cdot q_3^{'m_{j*}'}))^{1/\overline{h_1}'}$.

We can see that if $\tau = 0$, the game is perfect. Assume $\mathscr{A}_1$ makes q_K times partial key query and q_T times tag query; the game is performed successfully with the probability of $(1-\gamma)^{q_K+q_T}$. Therefore, if $\mathscr{A}_1$ wins the game with the probability ε, β can successfully output the result of g^{ab} with the probability $\varepsilon' \geq (1-\gamma)^{q_K+q_T} \cdot \gamma \cdot \varepsilon$. As known, CDH problem $(g, \mathbb{G}_1, g^a, g^b)$ is hard for the group $\mathbb{G}_1$, so our proposal is secure against $\mathscr{A}_1$. □

Lemma 2. *If the CDH problem $(g, \mathbb{G}_1, g^a, g^b)$ is hard for the group $\mathbb{G}_1$, our scheme is secure against $\mathscr{A}_2$.*

Proof. If the adversary $\mathscr{A}_2$ wins the game, a simulator β can be designed to solve the CDH hard problem resorting to $\mathscr{A}_2$. Let $(g, \mathbb{G}_1, g^a, g^b)$ to be one CDH instance, and β computes g^{ab} by following steps.

Setup: β picks a random number $x \in Z_q^*$ as the master private key. $\mathscr{B}$ gives $\mathscr{A}_2$ all the public parameters as well as the master private key x.

h_1 query: $\mathscr{B}$ makes a list $\text{tab}_1 = \{(\text{id}, \overline{h_1})\}$ for the h_1 query. If the user identity id queried exists in tab_1, $\mathscr{B}$ retrieves the row $(\text{id}, \overline{h_1})$ and responds the value $g^{\overline{h_1}}$ to $\mathscr{A}_2$. Otherwise, $\mathscr{B}$ selects a random number of $\overline{h_1^*} \in Z_q^*$, responds $g^{\overline{h_1^*}}$ to $\mathscr{A}_2$, and inserts $(\text{id}^*, \overline{h_1^*})$ to tab_1.

Secret value query: $\mathscr{A}_2$ can query the secret value for any user identity id^*. $\mathscr{B}$ makes a list $\text{tab}_2 = \{(\text{id}, pk, s)\}$ to trace the results for this query. If id^* existing in tab_2, $\mathscr{B}$ returns s to $\mathscr{A}_2$. Otherwise, $\mathscr{B}$ selects a random number $s^* \in Z_q^*$ and computes $\text{pk}^* = (g)^{1/s^*}$. $\mathscr{B}$ inserts the row (id^*, pk^*, s^*) to tab_2 and returns s^* to $\mathscr{A}_2$.

Public key query: $\mathscr{A}_2$ can query the public key for any user identity id^*. $\mathscr{B}$ searches id^* from tab_2. If id^* existing in tab_2, $\mathscr{B}$ responds pk^* to $\mathscr{A}_2$. Otherwise, $\mathscr{B}$ chooses a random value $s^* \in Z_q^*$ and computes $\text{pk}^* = (g^a)^{1/s^*}$. $\mathscr{B}$ inserts the row (id^*, pk^*, s^*) to tab_2 and returns pk^* to $\mathscr{A}_2$.

h_2 query: $\mathscr{A}_2$ can query the hash value of (Fid, j) at any time. For this query, β keeps a list tab_3 with tuple $(\text{Fid}, j, \overline{h_2})$. If the row $(\text{Fid}, j, *)$ exists in tab_3, β retrieves $\overline{h_2}$ and returns $(g^b)^{\overline{h_2}}$ to $\mathscr{A}_2$. Otherwise, β randomly chooses $\overline{h_2^*} \in Z_q^*$ and returns $(g^b)^{\overline{h_2}}$ to $\mathscr{A}_2$. β inserts a new row $(\text{Fid}, j, \overline{h_2^*})$ into tab_3.

h_3 query: $\mathscr{A}_2$ can query the hash value of (P_0, v^*) at any time. For this query, β keeps a list tab_4 with tuple $(P_0, v, q_3, \overline{h_3})$ and presets a special row $(P_0, V, g^{(a\overline{h_3})}, \overline{h_3})$. If the row $(P_0, v^*, *, *)$ exists in tab_4, β retrieves q_3^* and returns it to $\mathscr{A}_2$. Otherwise, β randomly chooses $\overline{h_3^*} \in \mathbb{G}_1$ and sets $q_3^* = g^{\overline{h_3^*}}$. β inserts a new row $(P_0, v*, q_3^*, \overline{h_3^*})$ into tab_4.

Tag query: for the tag query $(\text{Fid}, j, m_j, \text{id}, v)$, β gets the rows $(\text{id}, \overline{h_1})$, (id, pk, s), $(\text{Fid}, j, \overline{h_2})$, and $(P_0, v, q_3, \overline{h_3})$ from tab_1, tab_2, tab_3, and tab_4, respectively. Then, β computes the tag $T_{i,j} = (g^{x\overline{h_1}} \cdot g^{b\overline{h_2}} \cdot q_3^{m_j})^s$ and returns it to $\mathscr{A}_2$.

Forge: at last, $\mathscr{A}_2$ gives a forged tag $T'_{i*,j*}$ for block m'_{j*} with the identity id_{i*}'.and total block number v'. The block m'_{j*} has not be executed the tag query under such conditions before.

Analysis: if $\mathscr{A}_2$ wins the game, the following equation $e(T'_{i*,j*}, \text{PID}_{i*}) = e(h_1(\text{id}_{i*}'), g^x) \cdot e(h_2(\text{Fid}, j*) \cdot U^{m_{j*}'}, g)$ must hold according to equation (3). Then, β gets the row $(\text{id}_{i*}', \overline{h_1}')$, $(\text{id}_{i*}', pk', s')$, $(\text{Fid}, j^*, \overline{h_2}')$, and $(P_0, v', q_3', \overline{h_3}')$ from tab_1, tab_2, tab_3, and tab_4. If $v' \neq V$, β aborts and exits. Otherwise, β changes the equation above to $e(T'_{i*,j*}, g^{1/s'}) = e(g^{\overline{h_1'}}, g^x) \cdot e(g^{ab\overline{h_2'}\overline{h_3'}m_{j*}'}, g)$. We can compute that the result of given CDH instance is $g^{ab} = ((T'_{i*,j*})^{1/s'} / g^{x\overline{h_1}'})^{(1/(\overline{h_2}'\overline{h_3}'m_{j*}'))}$.

According to the analysis, if $v' = V$, β can successfully output the result of g^{ab}. Assume $\mathscr{A}_2$ makes q_K times h_3 query, and also, there are q_K rows in the tab_4. Thus, if $\mathscr{A}_2$ wins the game with the probability ε, β can get the value of g^{ab} with the probability (ε/q_K). Because CDH problem is hard for the group $\mathbb{G}_1$, our proposal is secure against for $\mathscr{A}_2$.

According to the Lemmas 1 and 2, our proposed scheme can resist both the adversaries of $\mathscr{A}_1$ and $\mathscr{A}_2$. Therefore, we can give Theorem 1 as □

Table 1: Comparison of computational cost.

Schemes	Tag generation	Proof generation	Proof audition	User privacy preserving
[31]	$2T_{\exp-G_1}$	$cT_{\exp-G_1}$	$3T_p + 2c \cdot T_{\exp-G_1}$	No
[36]	$2T_{\exp-G_1}$	$cT_{\exp-G_1}$	$(\lvert U\rvert + 2)T_p + (c + \lvert U\rvert) \cdot T_{\exp-G_1}$	No
[37]	$4T_{\exp-G_1}$	$cT_{\exp-G_1}$	$4T_p + (2c + 2) \cdot T_{\exp-G_1}$	No
Our scheme	$2T_{\exp-G_1}$	$(c + 1)T_p + 2cT_{\exp-G_1}$	$2T_p + (c + 1) \cdot T_{\exp-G_1}$	Yes

Theorem 1. *If the CDH problem is hard for the group $\mathbb{G}_1$, our proposed data integrity auditing scheme is secure in the random oracle model.*

Theorem 2. *TPA cannot reveal the identity of data uploader within the process of data auditing.*

Proof. From the audition algorithm of our scheme, it is not difficult to prove that TPA cannot know the data uploader of challenged data. First, the user's identity is stored by CSP privately, and no one knows the relation between data and user identity except CSP and users themselves. In the verification phase, TPA checks the proof by equation (6) without any information about user identity. Moreover, CSP also hides the user identity in the proof $\sigma_1 = \alpha \cdot \prod_{(m_l, T_{i,l}, \mathrm{id}_i) \in \Theta} h_1(\mathrm{id}_i)^{v_l}$ by random value α. Therefore, our scheme can guarantee the user privacy against TPA. □

6. Performance Analysis

6.1. Performance Evaluation. We summary the performance of our protocol from aspects of computational and communicational cost, which are shown as follows (Table 1).

Computational cost: let T_p, $T_{\exp-G_1}$, and $T_{\exp-G_2}$ represent the computational cost of pairing, exponentiation on $\mathbb{G}_1$, and exponentiation on $\mathbb{G}_2$, respectively. Others like hash function, addition, and multiplication on Z_q are omitted because they only incur negligible cost. It is easy to see that the algorithms such as Setup, PartialKeyGen, SecretValueGen, PublicKeyGen, and Challenge only need negligible cost, so we omit the performance analysis about these algorithms. The algorithm TagGen needs $2T_{\exp-G_1}$ for generating one tag. Thus, the computational cost for generating all tags is $2nT_{\exp-G_1}$. Proof algorithm is performed by CSP to generate proof which needs cost of $2cT_{\exp-G_1} + (c + 1)T_p$. The algorithm Audit is run by TPA, and it costs $2T_p + (c + 1) \cdot T_{\exp-G_1}$. Moreover, we compare the computational cost of our scheme with that in other three similar schemes in Table 1, in which $\lvert U\rvert$ is the count of group users.

From Table 1, we can get that the tag generation cost of our scheme is almost the same as that in [31, 36], which is much lower than that of [37]. In the proof generation step, our scheme has the highest cost than that of other three, that is, because our scheme does more work to hide the relationship of data and data uploader, so as to realize the user privacy preserving. We can see that only our scheme can preserve user privacy against TPA, while other three cannot. In the proof audition step, our scheme is the most efficient one compared with other three schemes. In summary, our scheme is computationally efficient.

Communicational cost: in our scheme, a tag is one element of $\mathbb{G}_1$, so the communication cost for data transfer form is $n\lvert\mathbb{G}_1\rvert + \lvert F\rvert + \lvert\mathrm{id}\rvert$, where $\lvert F\rvert$ denotes the size of outsourced data and $\lvert\mathrm{id}\rvert$ is the size of user identity. The size of each challenge is bounded of $4c + c\lvert Z_q\rvert$, and the proof size is $\lvert Z_q\rvert + \lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$.

6.2. Experiment Results. We implemented a prototype of our scheme with PBC library [43], which is based on the library of GMP [44]. Our experiments are executed in the Ubuntu Kylin-15.10 operating system with VMware workstation. We give 1 CPU and 1G Ram to the virtual machine and use the Lenovo laptop X270 as the host which installs the Win10 operation system with Core i5 CPU and 8G Ram. We choose the typical "Type A" elliptic curve supplied by PBC in our experiments. In order to accurately show the advantage of our scheme, we implement schemes in [31], [36], and [37] simultaneously.

We first make experiments to evaluate the efficiency of tag generation. We prepare 1000 randomly selected data blocks and run ten experiments with different number of tags. The results are shown in Figure 2. We can see that the computation cost increases linearly with the number of tags rising, which is consistent with the theoretical analysis. However, computing 1000 tags only costs about 9.8 seconds which is feasible.

Second, we make experiments to test the performance of proof generation. In this experiment, we simulate 100 different users and change the number of challenged blocks from 100 to 500 with total 1000 blocks. The experiment data are shown in Figure 3. From Figure 3, we can see that our scheme costs much more time than that of other three. The reason for this situation is analyzed before; specifically, we embed the relationship of challenged data and data uploader into the proof while other three schemes compute the proof only with data and tags without hiding the relationship. When checking the proof, TPA in other three schemes should use the data owner's public key which exposes the relationship of challenged data and the data owner.

The cost of proof audition is shown in Figure 4. The schemes in [31, 37] have the similar cost, the gap of which is very small. The cost of scheme in [36] is associated with the number of group users, so it has the most cost in the

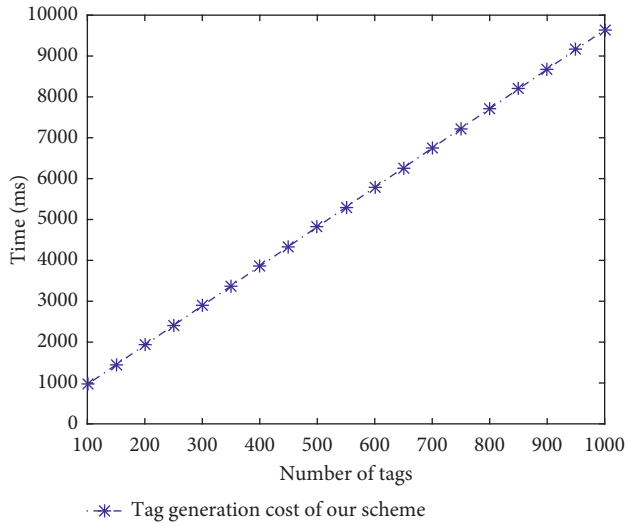

Figure 2: The cost of tag generation.

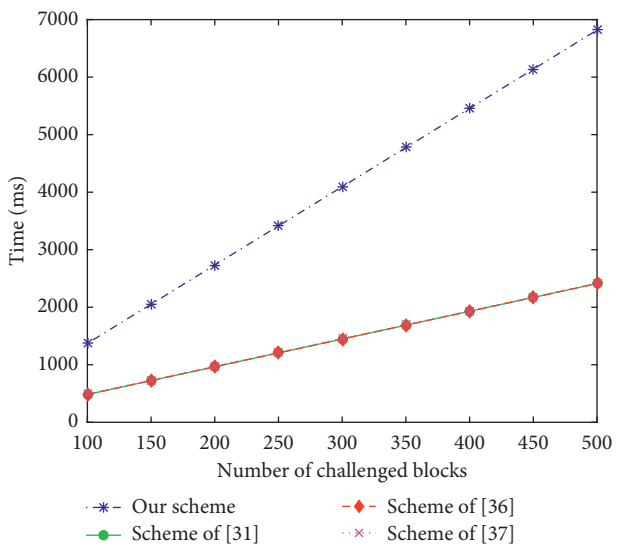

Figure 3: The cost of proof generation.

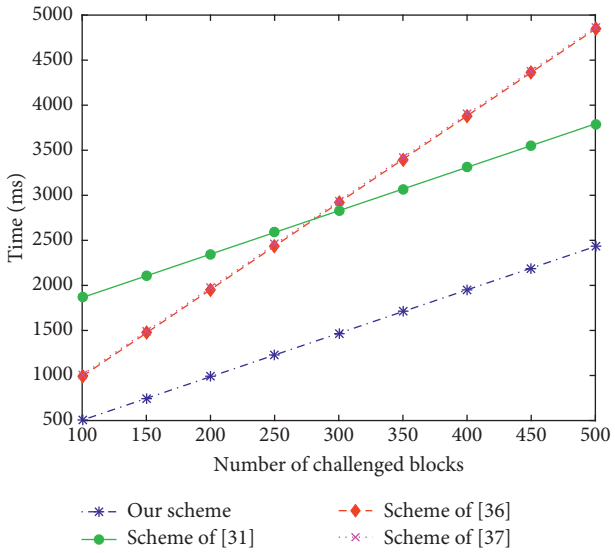

Figure 4: The cost of proof audition.

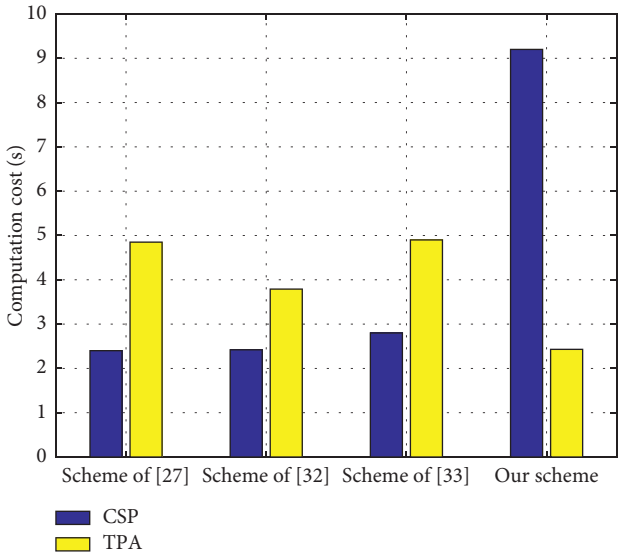

Figure 5: Computation cost of TPA and CSP.

beginning. However, with the number of challenged blocks increasing, the audition cost of [31, 37] exceeds that of the scheme [36]. Overall, our scheme is the most efficient, one in this step, which needs only 2.5 seconds for 500 challenged blocks.

It is well known that CSP has great computation ability, but TPA is usually a normal workstation or personal computer. Although our scheme costs more time when generating the proof in the experiments, it is done by CSP which makes the gap be negligible in real environment. However, the different of TPA in our experiments and in real environment is very small, so the advantage of proof audition in our scheme is the very important.

To improve the efficiency of the data integrity audition scheme, we can assign more workload to CSP but less to TPA. We summary the computation cost of CSP and TPA in the four schemes with 500 challenged blocks. The results are shown in Figure 5, from which we can see that our scheme assigns the most workload to CSP but the lightest workload to TPA. Thus, compared with recent researches, our scheme is efficient especially for TPA.

7. Conclusion

In this article, we propose a public certificateless PDP scheme for cloud storage. Our scheme not only inherits the advantages of certificateless cryptography but also has the merit of user identity privacy protection. With our scheme, TPA can audit the integrity of group shared data rightly without revealing the data uploader so as to preserve user's privacy. We formalize the security model of our scheme with

two types of adversaries and prove its security in the random oracle model. Experimental result demonstrates that our proposal is efficient.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Program for Scientific Research Foundation for Talented Scholars of Jinling Institute of Technology (JIT-B-202031), the Opening Foundation of Fujian Provincial Key Laboratory of Network Security and Cryptology Research Fund, Fujian Normal University (NSCL-KF2021-02), the National Natural Science Foundation of China (61902163), and the Key Program of National Key Research and Development Project "Cybersecurity" (2017YFB0802800).

References

[1] M. Ali, R. Dhamotharan, E. Khan et al., "SeDaSC: secure data sharing in clouds," *IEEE Systems Journal*, vol. 11, no. 2, pp. 1–10, 2015.

[2] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[3] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proceedings of the Conference on Hot Topics in Cloud Computing*, pp. 14–19, San Diego, CA, USA, June 2009.

[4] X. Yan, J. Cao, L. Sun, J. Zhou, S. Wang, and A. Song, "Accurate analytical-based multi-hop localization with low energy consumption for irregular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2021–2033, 2020.

[5] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: opportunities and challenges," *Information Sciences*, vol. 305, pp. 357–383, 2015.

[6] X. Yan, L. Sun, Z. Sun, J. Zhou, and A. Song, "Improved hop-based localisation algorithm for irregular networks," *IET Communications*, vol. 13, no. 5, pp. 520–527, 2019.

[7] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1214–1226, 2019.

[8] L. Chen, J. Li, Y. Lu, and Y. Zhang, "Adaptively secure certificate-based broadcast encryption and its application to cloud storage service," *Information Sciences*, vol. 538, pp. 273–289, 2020.

[9] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the fourteenth ACM Conference on Computer and Communications Security, ACM*, pp. 598–609, Alexandria, VA, USA, October 2007.

[10] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm'08)*, pp. 1–10, Istanbul Turkey, September 2008.

[11] A. Juels and B. S. Kaliski Jr., "PORs: proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS' 07)*, pp. 584–597, New York, NY, USA, October 2007.

[12] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proceedings of the 14th Annual International Conference on the Theory and Application of Cryptology and Information Security*, pp. 90–107, Melbourne, Australia, December 2008.

[13] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)*, pp. 213–222, Chicago, IL, USA, November 2009.

[14] H. Yan, J. Li, J. Han, and Y. Zhang, "A novel efficient remote data possession checking protocol in cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 78–88, 2017.

[15] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "MuR-DPA: top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2609–2622, 2015.

[16] J. Li, H. Yan, and Y. Zhang, "Efficient identity-based provable multi-copy data possession in multi-cloud storage," *IEEE Transactions on Cloud Computing*, p. 1, 2019.

[17] H. Wang, "Proxy provable data possession in public clouds," *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 551–559, 2013.

[18] H. Yan, J. Li, and Y. Zhang, "Remote data checking with a designated verifier in cloud storage," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1788–1797, 2020.

[19] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[20] W. Shen, G. Yang, J. Yu, H. Zhang, F. Kong, and R. Hao, "Remote data possession checking with privacy-preserving authenticators for cloud storage," *Future Generation Computer Systems*, vol. 76, pp. 136–145, 2017.

[21] B. Wang, B. Li, and H. Li, "Knox: privacy-preserving auditing for shared data with large groups in the cloud," in *Proceedings of 10th International Conference Applied Cryptography and Network Security (ACNS' 12)*, pp. 507–525, Singapore, June 2012.

[22] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," *Journal of Systems and Software*, vol. 113, pp. 130–139, 2016.

[23] B. Wang, H. Li, and M. Li, "Privacy-preserving public auditing for shared cloud data supporting group dynamics," in *Proceedings of IEEE International Conference Committee (ICC' 13)*, pp. 1946–1950, Budapest, Hungary, June 2013.

[24] G. Wu, Y. Mu, W. Susilo, and F. Guo, "Privacy-preserving cloud auditing with multiple uploaders," in *Proceedings of International Conference on Information Security Practice and Experience (ISPEC'06)*, pp. 224–237, Zhangjiajie, China, November 2016.

[25] B. Wang, B. Li, and H. Li, "Panda: public auditing for Shared data with efficient user revocation in the cloud," *IEEE*

Transactions on Services Computing, vol. 8, no. 1, pp. 92–106, 2015.

[26] H. Wang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Identity-based remote data possession checking in public clouds," *IET Information Security*, vol. 8, no. 2, pp. 114–121, 2014.

[27] Y. Yu, Y. Mu, J. Ni, J. Deng, and K. Huang, "Identity privacy-preserving public auditing with dynamic group for secure mobile cloud storage," in *Proceedings of 8th International Conference on Network and System Security (NSS' 14)*, pp. 28–40, Xi'an, China, October 2014.

[28] S. Tan and Y. Jia, "NaEPASC: a novel and efficient public auditing scheme for cloud data," *Journal of Zhejiang University Science C*, vol. 15, no. 9, pp. 794–804, 2014.

[29] X. Zhang, Y. Tang, H. Wang, C. Xu, Y. Miao, and H. Cheng, "Lattice-based proxy-oriented identity-based encryption with keyword search for cloud storage," *Information Sciences*, vol. 494, pp. 193–207, 2019.

[30] X. Zhang, J. Zhao, C. Xu, H. Li, H. Wang, and Y. Zhang, "CIPPPA: conditional identity privacy-preserving public auditing for cloud-based wbans against malicious auditors," *IEEE Transactions on Cloud Computing*, p. 1, 2019.

[31] B. Wang, B. Li, and H. Li, "Certificateless public auditing for data integrity in the cloud," in *Proceedings of IEEE Conference on Communications and Network Security (CNS)*, pp. 136–144, National Harbor, MD, USA, October 2013.

[32] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 64–73, 2018.

[33] Y. Liao, Y. Liang, A. W. Oyewole, and X. Nie, "Security analysis of a certificateless provable data possession scheme in cloud," *IEEE Access*, vol. 7, pp. 93259–93263, 2019.

[34] D. Kim and I. R. Jeong, "Certificateless public auditing protocol with constant verification time," *Security and Communication Networks*, vol. 201714 pages, Article ID 6758618, 2017.

[35] H. Yang, S. Jiang, W. Shen, and L. Zhou, "Certificateless provable group shared data possession with comprehensive privacy preservation for cloud storage," *Future Internet*, vol. 10, no. 6, 2018.

[36] J. Li, H. Yan, and Y. Zhang, "Certificateless public integrity checking of group shared data on cloud storage," *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 71–81, 2018.

[37] B. Kang, J. Wang, and D. Shao, "Certificateless public auditing with privacy preserving for cloud-assisted wireless body Area Networks," *Mobile Information Systems*, vol. 20175 pages, Article ID 2925465, 2017.

[38] Y. Ming and W. Shi, "Efficient privacy-preserving certificateless provable data possession scheme for cloud storage," *IEEE Access*, vol. 7, pp. 122091–122105, 2019.

[39] G. Wu, Y. Mu, W. Susilo, F. Guo, and F. Zhang, "Privacy-preserving certificateless cloud auditing with multiple users," *Wireless Personal Communications*, vol. 106, no. 3, pp. 1161–1182, 2019.

[40] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of the Annual International Cryptology Conference*, pp. 213–229, Santa Barbara, CA, USA, August 2001.

[41] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 452–473, Taipei, Taiwan, November 2003.

[42] H. Yan and W. Gui, "Efficient identity-based public integrity auditing of shared data in cloud storage with user privacy preserving," *IEEE Access*, vol. 9, pp. 45822–45831, 2021.

[43] "The pairing-based cryptography library (pbc)," 2020, https://crpto.stanford.edu/pbc/download.html.

[44] "The GNU multiple precision arithmetic library (GMP)," 2020, http://gmplib.org/.