IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 6, 2021, accepted March 6, 2021, date of publication March 17, 2021, date of current version March 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3066497

Efficient Identity-Based Public Integrity Auditing of Shared Data in Cloud Storage With User Privacy Preserving

HAO YAN[1] AND WENMING GUI[1,2]

[1]School of Network Security, Jinling Institute of Technology, Jiangsu 211169, China
[2]Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education, Jiangsu 210003, China

Corresponding author: Hao Yan (pxy_hao@163.com)

This work was supported in part by the Program for Scientific Research Foundation for Talented Scholars of Jinling Institute of Technology under Grant JIT-B-202031, and in part by the Open Research Fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education (Personalized music recommendation oriented by multimodal social network big data).

ABSTRACT Provable Data Possession (PDP) model provides an efficient means for people to audit the integrity of data stored in cloud storage. When sensitive data is shared among multiple users based on cloud storage, it is critical to preserve the anonymity of the data uploader against the auditor. That is, the auditor should not get data uploader's identity through the data audition. To address this problem, many PDP schemes with user identity privacy-persevering are proposed. However, most proposed schemes are designed based on PKI technique which suffers from big burden of certificate management. Moreover, data auditors in most proposed schemes bear heavy computation cost which results to the lower efficiency of the scheme. To overcome the shortcomings, we present a novel identity-based PDP protocol to audit efficiently the integrity of group shared data with uploader's privacy-preserving. Due to the inherent structural advantage of identity-based crypto mechanism, our PDP scheme is able to avoid the problem of certificate management. Different from previous works, our scheme ensures the relationship of the data and the data uploader in the phase of proof generation not the phase of integrity audition. Therefore, the data auditor does not know the relationship at all as well as the extract data uploader of the challenged data. At the same time, establishing the relationship by cloud server in proof generation step can reduce the computational cost of data auditor greatly. Furthermore, the relationship of data uploader and challenged data in the proof is randomized so as to strength the security of the scheme. All these efforts are made in our scheme to efficiently realize the anonymity protection of the data uploader. We give the detailed security proof of our scheme under the computational Diffie-Hellman assumption. Many experiments are performed to evaluate the efficiency of our scheme, the results show that our new scheme is efficient and feasible.

INDEX TERMS Cloud secure storage, identity-based cryptography, group data integrity checking, user privacy preserving, efficiency and security.

I. INTRODUCTION

Nowadays, explosive growth of data makes people bear big burden to store and manage data in local. To relieve the cost of data storage and management, more and more people rent cloud storage service and outsource the data to cloud servers. Furthermore, based on cloud storage, people are able to conveniently share data with each other and work as a team [1]–[3]. However, cloud service provider (CSP) is not fully trustworthy. The data stored in CSP may be corrupted or deleted due to accidental hardware errors, network exceptions, software bugs, or human mistakes [4]–[6]. To escape economic compensation and keep good reputation, CSP would not tell the truth to data user. Therefore, users need to periodically check whether the data in cloud storage server is kept well.

PDP model supplies user an efficient method to remotely verify the integrity of the data in cloud storage. PDP divides the outsourced data into many small data blocks and blinds one tag to each block. Since the tag contains the value of the data block, user can get the integrity status of the data block through checking the validity of the corresponding tag.

The associate editor coordinating the review of this manuscript and approving it for publication was SK Hafizul Islam.

Until now, several PDP schemes [7]–[37] with public verification have been proposed. Most PDP protocols focus just on checking the integrity of single data [7]–[21] that belongs to only one user. However, in real applications, sharing data among multiple users is a common situation, in which the shared data is able to be used by any one of the workgroup. Therefore, auditing the integrity of shared data becomes an attractive issue. When sensitive data is shared in a group, the data uploader's anonymity against third party auditor (TPA) must be preserved. Specifically, TPA should not get who the data uploader is after auditing the data integrity. That is, data integrity audition process should not reveal the confidential information of uploader's identity to TPA. Aim to this goal, Wang *et al.* [22] proposes a concrete PDP protocol with the notion of user privacy preserving for shared data. It resorts to group signature technique to keep user privacy private from the TPA. Following, several schemes [23]–[37] with user privacy preserving are proposed. However, most of these PDP schemes [23]–[32] are constructed on the PKI technique which suffers from certificate management problems such as certificate generation, distribution, revocation, re-new, update and verification.

To address this problem, some researchers utilize identity-based cryptography [38] and certificateless cryptography [39] to design PDP schemes [33]–[37] with user privacy preserving. Nevertheless, these schemes are not computationally efficient to apply in practical application. Therefore, it is necessary to present more efficient PDP scheme with user privacy preserving for cloud data audition.

A. MOTIVATION AND CONTRIBUTIONS

Most of previous PDP schemes only concentrate on verifying the integrity of personal data. However, sharing data with others based on cloud platform is a development trend. Because any user can save sensitive data on the cloud, the privacy of data uploader's identity should be guaranteed. That's to say, TPA can audit data integrity but can not distinguish the exact data uploader. For example, every person can report to the government about criminal behaviors through the open complaint platform. To prevent criminal from revenging the reporter, it's necessary to preserve the reporter's identity.

To address the problems above, the paper proposes a novel identity-based PDP scheme towards group shared data with user privacy protection. In our scheme, CSP presets the relationship between user identity and data block in integrity proof phase, so TPA can audit the correctness of the proof without knowing the relationship. As a result, user privacy is preserved against TPA. Moreover, detailed proof is given to prove the security of our proposal under defined security model. The evaluation results of experiments show that our PDP scheme is efficient and practical.

B. RELATED WORK

Ateniese *et al.* [7] firstly considered to check data integrity by PDP model and proposed two concrete schemes based on RSA algorithm. Similar to PDP, PoR model proposed by Juel and Kaliski *et al.* [9] has the function of remotely check data integrity too. To improve scheme efficiency, Shacham and Waters [8] developed a compact PoR scheme with shorter authentication tag. To support dynamic operations, Ateniese *et al.* [10] based on symmetric key encryption designed a more flexible PDP scheme, where data blocks can be appended, updated and deleted. Erway *et al.* [11] proposed a PDP protocol with full data block dynamic operations including data insertion. To improve dynamic operation efficiency, Yan *et al.* [12] realized a PDP scheme with the new data structure. Similarly, Shen *et al.* [13] designed another new data structure to realize data operations of their PDP scheme. To increase data durability, Liu *et al.* [14] proposed a multi-replicas data integrity checking protocol, which supported fully dynamic data updates. Wang [15] developed an integrity checking protocol for data on multi cloud servers. Li *et al.* [16] further considered a more complex environment that multi-copies stored in multi CSPs and constructed a concrete scheme to check the integrity of all copies for one time. To support delegation of data checking, Wang [17] proposed a proxy PDP scheme in which a commitment was used to authenticate the validity of auditor. Further, Yan *et al.* [18] strengthened the restriction of the verifier and proposed a verifier-designated PDP scheme. To preserve the data privacy, Wang *et al.* [19] proposed a notion of data privacy protection and designed a public auditable PDP scheme. To get rid of certificate management problem, Yu *et al.* [20] based on identity-based crypto [34] presented a PDP scheme with data privacy protection. Shen *et al.* [21] proposed a PDP protocol to guarantee the privacy of authenticators.

Wang *et al.* [22] proposed the first PDP model for data shared in group which utilized ring signature technique to generate tags so as to support public auditing and user privacy preserving. Wang *et al.* [23] proposed a new PDP scheme for shared data with user privacy preserving. Furthermore, the scheme in [23] also supported dynamic group which allowed user to join or leave the group at any time. Liu *et al.* [25] designed a PDP scheme based on broadcast encryption [24] supporting dynamic group. Wang *et al.* [26] considered the user revocation issue and proposed a PDP scheme which outsourced user revocation to CSP by proxy re-signature technique. Yang *et al.* [27] designed a PDP protocol for group data with user identity privacy and traceability. Wu *et al.* [28] developed a PDP scheme for data shared within multiple uploaders. Nayak and Tripathy [29] proposed a SEPDP scheme with data privacy preserving. They embedded the challenged data block to the proof as an exponent parameter so that TPA cannot recover the block from the proof. Moreover, they extended their scheme to support batch auditing and data dynamic. However, Yu and Hao [30] proved the scheme [29] was not secure to resist the forge attack of malicious CSP. Mara *et al.* [31] presented a CRUPA scheme to audit the data shared in a group. CRUPA made use of the concept of regression technique to resist the collusion attack of CSP and revoked users. Lu *et al.* [32] designed a data integrity verification mechanism for mobile terminals in

cloud computing. The scheme supported data privacy preserving and authorized access of the data. These schemes mentioned above mainly relied on the PKI technique which bears heavy burden for certificate management. To address the problem, Yu *et al.* [33] utilized the identity-base crypto to propose a PDP protocol with user privacy preserving in dynamic group. However, this scheme was only suitable for devices with limited computational ability. To avoid certificate management and key escrow, Li *et al.* [34] proposed a PDP scheme of group shared data based on certificateless cryptography. However, the scheme lost the user privacy preservation feature. Similarly, Yang *et al.* [35] presented a scheme of shared data based on certificateless cryptography too. Although the scheme claimed that it was able to guarantee user identity, unfortunately, TPA can get the relationship of data and the public keys in the verification phase. Thus, it did not really realize user privacy preserving. Wu *et al.* [36] presented a new PDP scheme with user privacy protection, but the communication and computation overheads of the scheme were too heavy especially in the challenge phase.

II. PRELIMINARIES

We first review some preliminary cryptography knowledge throughout this paper.

A. BILINEAR MAPS

Assume 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 $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map with the properties as follow.

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

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

(iii) Non-degeneracy: $\exists u, v \in \mathbb{G}_1$ so that $e(u, v) \neq 1_{\mathbb{G}_2}$.

B. ASSUMPTION

Definition 1. Computational Diffie-Hellman assumption: Let g be a generator of 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 $\mathcal{A}$ (probabilistic polynomial time, PPT), the probability for $\mathcal{A}$ to solve this problem (CDH) is negligible, which can be denoted as:

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

III. SYSTEM MODEL AND SECURITY MODEL

A. SYSTEM MODEL

There are four participants in our scheme: key generation center, CSP, users and TPA.

(1) key generation center (KGC) generates the private keys for all users. We assume the keys are transmitted by secure channel.

(2) CSP maintains user's data and generates the proofs for data integrity challenge from TPA.

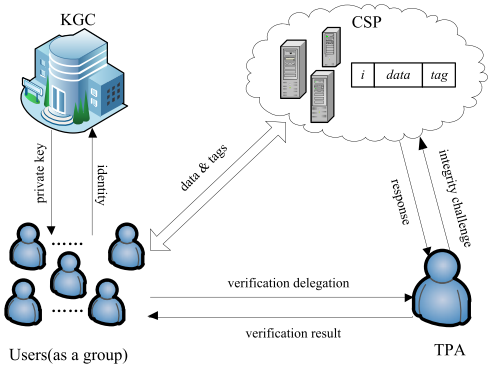

FIGURE 1. System model of our scheme.

(3) users generate tags for their data and outsource the data with tags to CSP. Here, all users share their data to each other in a group.

(4) TPA audits the integrity of data shared within a group. TPA first sends an integrity challenge to CSP and gets a proof from CSP. Then TPA validates the rightness of the proof and reports the checking result to users. Assume TPA is able to honestly execute the audition process.

The system model is illustrated in Figure. 1. It assumed that CSP is semi-trusted. Namely, it can execute audition protocol honestly, but lies to TPA when data is broken. TPA is assumed to be honest-but-curious, that is, TPA performs the audition for data integrity honestly and responds the real audition result to users, but it is curious to reveal the identity of data uploader.

B. DEFINITION OF OUR SCHEME

A public identity-based auditing scheme for shared data supporting user privacy preserving consist of six algorithms *Setup*, *Extract*, *TagGen*, *Challenge*, *Proof* and *Audit* which are described as below:

$Setup(1^k) \rightarrow (pp, msk)$ With the security parameter k, this algorithm outputs the public parameter pp and the master key msk.

$Extract(ID_j, msk) \rightarrow sk_{ID_j}$: This algorithm outputs user secret key sk_{ID_j} with user's identity $ID_j \in \{0, 1\}^*$ and the master key msk.

$TagGen(sk_{ID_j}, m_i) \rightarrow T_{i,j}$: The algorithm generates one authentication tag for each data block. It inputs user's secret key sk_{ID_j} and the data m_i, outputs tag $T_{i,j}$.

$Challenge(Fid) \rightarrow chal$ This algorithm is performed by TPA to generate a data integrity challenge $chal$ for data named Fid.

$Proof(F, T, chal) \rightarrow P$ The algorithm generates the data integrity proof P for $chal$. It takes the inputs of challenged data F, tags collection T and challenge $chal$.

$Audit(chal, P, Fid) \rightarrow \{0, 1\}$ This algorithm is used to audit the rightness of integrity proof. It takes the inputs of challenge $chal$, proof P and the name Fid. It returns '1' if P passed the audition, else returns '0'.

We give the detailed process flow of our scheme: KGC runs the *Setup* to initialize the system and runs the *Extract*

to generate the private keys for all users. Users in the group prepare their data and compute all the tags of the data by *TagGen*. They outsource the data and the tags to CSP. TPA runs *Challenge* to send an integrity challenge request to CSP. CSP generates an integrity proof for the challenge request by *Proof* and submits the proof to TPA. TPA runs *Audit* to check the correctness of the proof and return the checking result to users.

C. SECURITY MODEL

A public identity-based auditing scheme for group shared data supporting user privacy preserving should achieve three security features: completeness, soundness and user privacy protection against TPA. Completeness means the integrity of shared data should be audited rightly when CSP and TPA execute the protocol honestly. Soundness means when data is broken the scheme can resist CSP cheating TPA by forging the proof. Namely, if CSP doesn't maintain the challenged data blocks, it can't output correct data integrity proof. User privacy preserving means the data uploader's identity should be guaranteed against the auditor. That is to say, TPA should not obtain data uploader's identity during the procedure of data integrity auditing.

Completeness of the scheme is defined as:

Definition 2: A public identity-based auditing scheme for data shared with multi-users is effective, if the equation $Audit(chal, Proof(F, T, chal), Fid) = 1$ always holds.

Soundness of the scheme can be captured by a game. The game involves an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. We describe the game as below:

Setup Phase: $\mathcal{C}$ runs *Setup* algorithm to set the public parameter *pp* and the master key *msk*. $\mathcal{C}$ stores *msk* and gives *pp* to $\mathcal{A}$.

Queries Phase:$\mathcal{A}$ makes three types of query to $\mathcal{C}$ for polynomial times. $\mathcal{C}$ responds the query results to $\mathcal{A}$.

(a) Hash Query. adversary $\mathcal{A}$ queries the hash values of any hash function in the scheme. $\mathcal{C}$ replies the hash values to $\mathcal{A}$.

(b) Private-Key Query. $\mathcal{A}$ can query any user's private key with the identity ID_j. $\mathcal{C}$ calculates the private key sk_{ID_j} by the algorithm *Extract* and returns the key to $\mathcal{A}$.

(c) Tag Query. adversary $\mathcal{A}$ can send randomly selected blocks to $\mathcal{C}$ and query their tags generated by any user in the group. $\mathcal{C}$ runs algorithm *TagGen* to generate the tag of the queried block and sends the tag back to $\mathcal{A}$. If $\mathcal{C}$ does not have user's private key, it can compute the key by *Extract* algorithm.

Challenge Phase: $\mathcal{C}$ runs *Challenge* to get a challenge *chal* and submits it to $\mathcal{A}$. Noted that at least one block in *chal* has not been queried by $\mathcal{A}$. $\mathcal{C}$ asks $\mathcal{A}$ to respond a proof for *chal*.

Forge Phase: Finally, $\mathcal{A}$ submits a proof P to $\mathcal{C}$ for the challenge *chal*. If P passes the audition, $\mathcal{A}$ wins the game.

*Definition 3:*A public identity-based auditing scheme for group shared data supporting user privacy preserving is secure, if any adversary $\mathcal{A}$ wins the game above only with negligible probability.

User privacy preserving is an important security feature of the scheme. The setting of our scheme is that multiple users share data with each other in a group and each one can upload data to the group. Since the data is sensitive and crucial, data uploader prefers to keep anonymous against TPA. However, an honest-but-curious TPA tries to distinguish the identity of data uploader during data verification process. It may result to the user information leakage which brings security threaten to data uploader. Thus, the scheme should guarantee data uploader's anonymity against TPA.

Definition 4: A public identity-based auditing scheme for shared data is user privacy-preserving, if TPA can not reveal the identity of data uploader within the procedure of data audition.

IV. CONCRETE CONSTRUCTION OF OUR SCHEME

We show the detailed construction of our identity-based auditing scheme for group shared data, which realizes public audition and user privacy protection.

Suppose U users work together as a team. Each user in the team is denoted by $u_j\,(1 \le j \le U)$ whose identity is ID_j. The team deals with the data F which is split into n blocks. Therefore, the data F can be represented as $\{m_i \mid 1 \le i \le n\}$, where i is the block index. The symbol $T_{i,j}$ is a block tag generated by the user u_j for the block m_i. The algorithms in our scheme are defined as follow.

$Setup(1^k) \to (pp, msk)$: KGC selects a big random prime number q with $|q| = k$ where k is the security parameter. Select two cyclic multiplicative groups $\mathbb{G}_1$ and $\mathbb{G}_2$with order q. $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear map on $\mathbb{G}_1$ and $\mathbb{G}_2$. Choose a generator g of $\mathbb{G}_1$ and two different hash functions H_1 and H_2 which are defined as:

$$H_1 : \{0,1\}^* \to \mathbb{G}_1$$
$$H_2 : \{0,1\}^* \to \mathbb{G}_1$$

Choose a pseudo-random function ϕ and a pseudo-random permutation π:

$$\phi : Z_q^* \times Z_q^* \to Z_q^*$$
$$\pi : Z_q^* \times \{1, \cdots, n\} \to \{1, \cdots, n\}$$

Then, KGC randomly selects two values: $s \in Z_q^*$, $u \in \mathbb{G}_1$ and sets the master secret key $msk = s$, the master public key $P_0 = g^s$. Thus, the system parameter is $pp = (q, g, \mathbb{G}_1, \mathbb{G}_2, u, e, P_0, H_1, H_2, \phi, \pi)$.

$Extract(ID_j, msk) \to sk_j$: on receiving the identity ID_j of the user u_j, KGC computes $sk_j = H_1(ID_j)^s$ as u_j's private key and sends it to the user u_j by secure channel.

$TagGen(sk_j, m_i) \to T_{i,j}$: each user can run this algorithm to compute the tag for any block. Take u_j as the example, u_j selects a random value $\lambda_j \in Z_q^*$ and generates the tag for the block m_i by the equation (1),

$$T_{i,j} = (sk_{ID_j} \cdot H_2(Fid||i) \cdot u^{m_i})^{1/\lambda_j} \tag{1}$$

Here, *Fid* is the unique identification of the data F. After getting the tag $T_{i,j}$, u_j chooses a secure signature scheme

SIG (such as BLS [36]) to compute the signature $\mu_j = SIG(R_j||ID_j)$, in which $R_j = g^{\lambda_j}$. Finally, u_j uploads $(m_i, T_{i,j}, ID_j, R_j, \mu_j)$ to CSP. Note that the values (ID_j, R_j, μ_j) only need to be uploaded once, because they are bound with the user u_j and keep unchanged in the system. When received the $(m_i, T_{i,j}, ID_j, R_j, \mu_j)$ from user, CSP first verifies the correctness of μ_j by the signature scheme *SIG*. If μ_j is invalid, CSP drops the data and notifies the user u_j. Otherwise, CSP validates the rightness of the tag by the equation (2):

$$e(T_{i,j}, R_j) = e(H_1(ID_j), P_0) \cdot e(H_2(Fid||i) \cdot u^{m_i}, g) \quad (2)$$

It can be confirmed as follow:

$$\begin{aligned} e(T_{i,j}, R_j) &= e((H_1(ID_j)^s \cdot H_2(Fid||i) \cdot u^{m_i})^{1/\lambda_j}, g^{\lambda_j}) \\ &= e(H_1(ID_j)^s, g) \cdot e(H_2(Fid||i) \cdot u^{m_i}, g) \\ &= e(H_1(ID_j), P_0) \cdot e(H_2(Fid||i) \cdot u^{m_i}, g) \end{aligned}$$

Challenge$(Fid) \rightarrow chal$: TPA runs this algorithm to challenge the integrity of data named *Fid*. TPA first sets the number of challenged blocks c and then randomly chooses two values $k_1, k_2 \in Z_q^*$. TPA submits the challenge request $chal = (c, k_1, k_2)$ to CSP.

Proof$(F, T, chal) \rightarrow P$: with $chal = (c, k_1, k_2)$ received from TPA, CSP randomly selects $h \in \mathbb{G}_1$ and computes the challenge set $C = \{(v_l, a_l) \mid 1 \le l \le c\}$ where $v_l = \pi(k_1, l)$, $a_l = \phi(k_2, l)$. Here,$\{v_l \mid 1 \le l \le c\}$ denotes the set of indexes of challenged blocks, $\{a_l \mid 1 \le l \le c\}$ denotes the set of random parameters. With the set $\{v_l \mid 1 \le l \le c\}$, CSP finds all the corresponding data uploader's identity ID_{j,v_l} from $(m_{v_l}, T_{v_l,j}, ID_{j,v_l}, R_j, \mu_j)$. Then, CSP computes:

$$\begin{aligned} \sigma_1 &= h \cdot \prod_{(v_l,a_l)\in C} H_1(ID_{j,v_l})^{a_l}, \\ \sigma_2 &= e(h, P_0) \cdot \prod_{(v_l,a_l)\in C} e(T_{v_l,j}^{a_l}, R_j), \\ M &= \sum_{(v_l,a_l)\in C} a_l m_{v_l}. \end{aligned}$$

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

Audit$(chal, P, Fid) \rightarrow \{0, 1\}$: After receiving P, TPA computes the challenge set $C = \{(v_l, a_l) \mid 1 \le l \le c\}$, where $v_l = \pi(k_1, l)$ and $a_l = \phi(k_2, l)$. TPA checks the equation (3):

$$\sigma_2 = e(\sigma_1, P_0) \cdot e(\prod_{(v_l,a_l)\in C} H_2(Fid||v_l)^{a_l} \cdot u^M, g) \quad (3)$$

If it holds, returns 1, otherwise returns 0.

V. SECURITY PROOF

A. COMPLETENESS PROOF

The completeness of the scheme can be proved as following:

$$\begin{aligned} &\sigma_2 \\ &= e(h, P_0) \cdot \prod_{(v_l,a_l)\in C} e(T_{v_l,j}^{a_l}, g^{\lambda_j}) \\ &= e(h, P_0) \cdot \prod_{(vl,al)\in C} e((H_1(ID_{j,v_l})^s \cdot H_2(Fid||v_l) \cdot u^{m_{v_l}})^{\frac{a_l}{\lambda_j}}, g^{\lambda_j}) \\ &= e(h, P_0) \cdot e(\prod_{(v_l,a_l)\in C} H_1(ID_{j,v_l})^{a_l}, g^s) \cdot \\ &\quad \times e(\prod_{(v_l,a_l)\in C} H_2(Fid||v_l)^{a_l} \cdot u^{\sum_{(v_l,a_l)\in C} a_l m_{v_l}}, g) \\ &= e(\sigma_1, P_0) \cdot e(\prod_{(v_l,a_l)\in C} H_2(Fid||v_l)^{a_l} \cdot u^M, g) \end{aligned}$$

B. SOUNDNESS PROOF

The soundness of our scheme can be proved in two steps. First, we prove the tag of any block can't be forged by CSP no matter who is the tag generator. Second, we prove the integrity proof can not be forged by CSP no matter what the challenge request is.

Theorem 1: The CDH problem can be broken with the probability $\varepsilon' \ge \varepsilon/((q_k + q_T) \cdot 2e)$ in the time $t' \le t + O(q_{H_1} + q_k + q_{H_2} + q_T)$, if there exists a PPT adversary wins the security game with advantage ε in time t, for at most $q_{H_1}, q_{H_2}, q_k, q_T$ times of H_1-Query, H_2-Query, PrivateKey-Query and Tag-Query respectively.

Proof: Assume the PPT adversary $\mathcal{A}$ wins the security game, we can get a simulator $\mathcal{B}$ to solve the CDH problem resorting to $\mathcal{A}$. Let $(g, \mathbb{G}_1, g^a, g^b)$ to be one CDH instance, $\mathcal{B}$ computes g^{ab} by following steps.

Setup: $\mathcal{B}$ sets the master public key $P_0 = g^a$, which means the master private key $msk = a$. Note that a is unknown to $\mathcal{B}$. $\mathcal{B}$ randomly selects public parameters $\lambda \in Z_q^*$, $u \in \mathbb{G}_1$ and sets $R = g^{\lambda}$. Then $\mathcal{B}$ gives $\mathcal{A}$ all the public parameters and the value R.

H_1-Query: $\mathcal{A}$ adaptively queries the hash value of any identity ID^*. $\mathcal{B}$ keeps a table $L_1 = \{(ID, h_1, Q_1, \tau)\}$ for the H_1-Query. If L_1 contains the row $(ID^*, *, *, *)$, $\mathcal{B}$ gets the row $(ID^*, h_1^*, Q_1^*, \tau^*)$ from L_1 and responds Q_1^* to $\mathcal{A}$. Otherwise, $\mathcal{B}$ randomly chooses a number $h_1^* \in Z_q^*$. Then $\mathcal{B}$ 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 $\tau = 1$, $\mathcal{B}$ sets $Q_1^* = (g^b)^{h_1^*}$. Otherwise, $\mathcal{B}$ computes $Q_1^* = g^{h_1^*}$. $\mathcal{B}$ responds Q_1^* to $\mathcal{A}$ and appends a new row $(ID^*, h_1^*, Q_1^*, \tau)$ into table L_1.

PrivateKey-Query: $\mathcal{A}$ sends any identity ID^* to $\mathcal{B}$ for querying the private key. $\mathcal{B}$ searches the row $(ID^*, h_1^*, Q_1^*, \tau^*)$ with ID^* from L_1. If it doesn't exist, $\mathcal{B}$ gets it by making H_1-Query for the $(ID^*$. When obtaining the row $(ID^*, h_1^*, Q_1^*, \tau^*)$, $\mathcal{B}$ checks the value τ^*. If $\tau^* = 1$, $\mathcal{B}$ aborts and exits the game. Otherwise, $\mathcal{B}$ computes $(Q_1^*)^a = (g^{h_1^*})^a = (g^a)^{h_1^*}$ and returns it to $\mathcal{A}$.

H_2 -Query: $\mathcal{A}$ can query the hash value of (Fid, i) at any time. For this query, $\mathcal{B}$ keeps a list L_2 with tuple (Fid, i, Q_2). If the row $(Fid, i, *)$ exists in L_2, $\mathcal{B}$ retrieves Q_2 and returns it to $\mathcal{A}$. Otherwise, $\mathcal{B}$ randomly chooses $Q_2' \in \mathbb{G}_1$ and returns Q_2' to $\mathcal{A}$. $\mathcal{B}$ inserts a new row (Fid, i, Q_2') into L_2.

Tag-Query: For this query, $\mathcal{B}$ gets the row (ID, h_1, Q_1, τ) from table L_1 and gets (Fid, i, Q_2) from table L_2. If not exist,

$\mathcal{B}$ can get them by H_1-Query and H_2-Query. If $\tau = 1, \mathcal{B}$ aborts and exits. Otherwise, $\mathcal{B}$ computes the tag: $T_{i,j} = ((g^a)^{h_1} \cdot Q_2 \cdot u^{m_i})^{1/\lambda}$.

Forge: At last, $\mathcal{A}$ gives a forged tag T'_{i^*,j^*} for block m'_{i^*} with the identity ID'_{j^*}. The block m'_{i^*} has not be executed the Tag-Query under such conditions before.

Analysis: It is easy to see that if $\mathcal{A}$ wins the game, the values $(m'_{i^*}, T'_{i^*,j^*}, ID'_{j^*})$ have to satisfy the equation (2). Then, we can get the equation (4):

$$e(T'_{i^*,j^*}, R) = e(H_1(ID'_{j^*}), g^a) \cdot e(H_2(Fid||i^*) \cdot u^{m'_{i^*}}, g) \tag{4}$$

To compute the value of g^{ab}, $\mathcal{B}$ first searches the row $(ID'_{j^*}, h'_1, Q'_1, \tau')$ from L_1. If $\tau' = 0$, $\mathcal{B}$ outputs "Fail" and exits the game. Otherwise, $\mathcal{B}$ continues to find the row (Fid, i^*, Q'_2) from L_2. Based on these values, the equation (4) can be changed to:

$$\begin{aligned} e(T_{i^*,j^*}{}', R) &= e(g^{bh'_1}, g^a) \cdot e(Q'_2 \cdot u^{m'_{i^*}}, g) \\ &= e(g^{abh'_1}, g) \cdot e(Q'_2 \cdot u^{m'_{i^*}}, g) \\ &= e(g^{abh'_1} \cdot Q'_2 \cdot u^{m'_{i^*}}, g) \end{aligned}$$

i.e., $e((T_{i^*,j^*}{}')^{\lambda}, g) = e(g^{abh'_1} \cdot Q'_2 \cdot u^{m'_{i*}}, g)$. Therefore, we can compute the result of given CDH instance:

$$g^{ab} = \left(\frac{(T_{i^*,j^*}{}')^{\lambda}}{Q_2{}' \cdot u^{m_{i^*}{}'}} \right)^{1/h'_1}.$$

According to the analysis, if $\tau = 1, \mathcal{B}$ outputs "Fail" and exists the game. Otherwise, the game is perfect. Therefore, the probability that $\mathcal{B}$ perfectly playing the game with $\mathcal{A}$ without abortion is higher than $(1-\gamma)^{q_k+q_T}$. As a result, $\mathcal{B}$ can successfully output the result of g^{ab} with the probability $\varepsilon' \geq \varepsilon \cdot \gamma \cdot (1-\gamma)^{(q_k+q_T)} \geq \varepsilon / ((q_k+q_T) \cdot 2e)$. The time cost of this process is $t' \leq t + O(q_{H_1} + q_k + q_{H_2} + q_T)$.

Theorem 2: If all hash functions in the scheme are collision-resistance, CSP generates the forged proof to cheat the TPA only with negligible probability.

Proof: The beforehand procedures are the same as that in the proof of 'Theorem 1'.

Suppose $chal = (c, k_1, k_2)$ is the challenge request to $\mathcal{A}$. $\mathcal{A}$ outputs a forged proof $P' = (\sigma'_1, \sigma'_2, M')$ which passes the audition.

Analysis: For the challenge $chal = (c, k_1, k_2)$, we can compute all the indexes of the challenged block $v_l = \pi(k_1, l)$ $(1 \leq l \leq c)$ and all the random parameters $a_l = \phi(k_2, l)(1 \leq l \leq c)$. Assume the forged proof is $P' = (\sigma'_1, \sigma'_2, M')$ where

$$\begin{aligned} \sigma_1{}' &= h \cdot \prod_{(v_l,a_l)\in C} H_1(ID_{j,v_l})^{a_l} \\ \sigma'_2 &= e(h, P_0) \cdot \prod_{(v_l,a_l)\in C} e(T'_{v_l,j}{}^{a_l}, R_j) \\ M' &= \sum_{(v_l,a_l)\in C} a_l m'_{v_l} \end{aligned}$$

Because the forged proof P' can pass the audition, P' has to satisfy the equation (3). Then we can get the equation (5):

$$\sigma'_2 = e(\sigma'_1, P_0) \cdot e(\prod_{(v_l,a_l)\in C} H_2(Fid||v_l)^{a_l} \cdot u^{M'}, g) \tag{5}$$

We assume the true proof for challenge $chal = (c, k_1, k_2)$ is $P = (\sigma_1, \sigma_2, M)$, where

$$\begin{aligned} \sigma_1 &= h \cdot \prod_{(v_l,a_l)\in C} H_1(ID_{j,v_l})^{a_l}, \\ \sigma_2 &= e(h, P_0) \cdot \prod_{(v_l,a_l)\in C} e(T_{v_l,j}^{a_l}, R_j) \\ M &= \sum_{(v_l,a_l)\in C} a_l m_{v_l} \end{aligned}$$

σ_1 and σ'_1 are computed with user identity regardless the block and tag, so it is easy to get $\sigma_1 = \sigma'_1$. Moreover, because P passes the audition, we can also get the equation (6):

$$\sigma_2 = e(\sigma_1, P_0) \cdot e(\prod_{(v_l,a_l)\in C} H_2(Fid||v_l)^{a_l} \cdot u^{M}, g) \tag{6}$$

Compared with the equations (5) and (6), we can see that if $M = M'$, then $\sigma_2 = \sigma'_2$. It means $P = P'$ which is contrast to the assumption. Therefore, M' must be not equal to M. Under this condition, we consider two cases: $\sigma_2 = \sigma'_2$ and $\sigma_2 \neq \sigma'_2$. If $\sigma_2 \neq \sigma'_2$, we consider the extreme situation that there is only one challenged block in the forged proof. This means the adversary can forge the tag for single block, which is contrast to the 'Theorem 1'. Then, we consider $\sigma_2 = \sigma'_2$. If $\sigma_2 = \sigma'_2$, according to the equations (5) and (6), we can get $u^M = u^{M'}$, i.e. $u^{\sum_{(v_l,a_l)\in C} a_l m_{v_l}} = u^{\sum_{(v_l,a_l)\in C} a_l m'_{v_l}}$, i.e., $\sum_{(v_l,a_l)\in C} a_l(m_{v_l} - m'_{v_l}) = 0$. It means each $m_{v_l} = m'_{v_l}$. However, it is contrast to our assumption of $M \neq M'$. Hence, the theorem 2 is proved.

C. PRIVACY PRESERVING

Theorem 3: TPA cannot get the identity of data uploader within the process of data auditing.

Proof: Look into the complete procedure of data integrity auditing carefully, it is not difficult to prove that TPA can not know the data uploader of challenged data. First, the user's identity is stored by CSP privately, no one knows the relation between data and user identity except CSP and user himself. When auditing the data, TPA sends a challenge to CSP, which contains no information about user. In audition phase, TPA checks the correctness of the proof by equation (3), which does not refer to user identity either. Moreover, CSP hides the user identity in the proof $\sigma_1 = h \cdot \prod_{(v_l,a_l)\in C} H_1(ID_{j,v_l})^{a_l}$ by random value h. Even there only one user identity in σ_1, TPA cannot obtain the user identity either. Therefore, our scheme can guarantee the user privacy against TPA.

D. PROBABILITY OF MISBEHAVIOR DETECTION

Our scheme adopts the random sampling method to detect the misbehavior of CSP which reduces the workload of

TABLE 1. Comparison of computational cost.

schemes	tag-gen	challenge	proof-gen	proof-verification
CL-PGSDP	$2T_{\exp\text{-}G_1}$	$cT_p + cT_{exp\text{-}G_2} + T_{exp\text{-}G_1}$	$T_p + (c+1)\cdot T_{exp\text{-}G_1} + c\cdot T_{exp\text{-}G_2}$	$c\cdot T_p + 2c\cdot T_{exp\text{-}G_1}$
CLCA	$2T_{\exp\text{-}G_1}$	$2U\cdot T_{\exp\text{-}G_1}$	$2c\cdot T_p + 2c\cdot T_{exp\text{-}G_2}$	$2T_p + (c+2)\cdot T_{exp\text{-}G_1} + T_{exp\text{-}G_2}$
ACAMU	$2T_{\exp\text{-}G_1}$	$(U+1)\cdot T_{\exp\text{-}G_1}$	$(2U+c)\cdot T_p + c\cdot T_{exp\text{-}G_1}$	$T_p + (c+2)\cdot T_{exp\text{-}G_1}$
Our scheme	$2T_{\exp\text{-}G_1}$	negligible	$(c+1)\cdot T_p + 2c\cdot T_{\exp\text{-}G_1}$	$2T_p + (c+1)\cdot T_{exp\text{-}G_1}$

TABLE 2. Comparison of communicational cost.

schemes	Tag size	Challenge size	Proof size
CL-PGSDP	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_1\rvert + c\cdot\lvert\mathbb{G}_2\rvert + 2c\cdot\lvert Z_q\rvert$	l
CLCA	$\lvert\mathbb{G}_1\rvert$	$2U\cdot\lvert\mathbb{G}_1\rvert + 2c\cdot\lvert Z_q\rvert$	$\lvert\mathbb{G}_2\rvert + \lvert Z_q\rvert$
ACAMU	$\lvert\mathbb{G}_1\rvert$	$(U+2)\cdot\lvert\mathbb{G}_1\rvert + 2c\cdot\lvert Z_q\rvert$	$\lvert\mathbb{G}_2\rvert + \lvert Z_q\rvert$
Our scheme	$\lvert\mathbb{G}_1\rvert$	$3\lvert Z_q\rvert$	$\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert + \lvert Z_q\rvert$

TPA. Assume user data is divided into n blcoks which are outsourced in CSP. With the challenge $chal = (c, k_1, k_2)$, CSP randomly selects c different blocks decided by the pseudo-random permutation π. Assume that CSP modifies c_1 blocks out of n blocks, so the percentage of tampered block is $P_t = c_1/n$. To detect the misbehavior, it is required that at least one tampered block is selected by CSP out of n blocks. Therefore, the probability of misbehavior detection is:

$$P = 1 - \frac{n-c_1}{n}\cdot\frac{n-c_1-1}{n-1}\cdots\frac{n-c_1-c+1}{n-c+1}$$

Obviously, we can get bigger detection probability when we increase the number of challenged blocks. The Figure 2 demonstrates the result of the detection probability with different number of challenged blocks. In this experiment we divide user data to 100000 blocks and set the P_t to 0.5%, 1%, 2% and 3% respectively. The number of challenged blocks increases form 100 to 1000 for each P_t. From the Figure 2, we can see if $P_t = 1\%$, we only need to challenge about 400 blocks to achieve $P > 98\%$. For $P_t = 2\%$, 180 blocks are enough to achieve $P > 98\%$. Thus, our scheme can efficiently detect the misbehavior of CSP by randomly sampling a few blocks.

VI. PERFORMANCE ANALYSIS

A. PERFORMANCE EVALUATION

We summary the performance of our protocol from aspects of computational and communicational cost, which are shown as follows.

Computational Cost: Let T_p, $T_{\exp-G_1}$, $T_{\exp-G_2}$ 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 is omitted. Suppose the data has n blocks in total, each challenge refers to c blocks. *Extract* algorithm needs only one $T_{\exp-G_1}$ operation. The algorithm *TagGen* needs $2T_{\exp-G_1}$ for generating one tag. Thus, the computational cost for generating all n tags is $2nT_{\exp-G_1}$. The *Challenge* algorithm only selects two values, it causes negligible cost. *Proof* algorithm is performed to generate proofs which needs cost of $2cT_{\exp-G_1} + (c+1)T_p$. To audit data integrity, the TPA needs to run the algorithm *Audit*, which costs $2T_p + (c+1)T_{\exp-G_1}$. Moreover, we compare our scheme with three similar schemes: ACAMU [28], CL-PGSDP [35] and CLCA [36] in terms of computational cost in Table 1, in which U is the number of group users.

The Table 1 shows that in tag generation phase our scheme has the same cost as others. In challenge phase, our scheme only needs negligible cost while other three schemes have great cost. In proof generation, the computational cost of our scheme is a little bigger than CL-PGSDP, but better than other two schemes. In proof verification phase, our scheme has the best performance. In summary, our scheme is computationally efficient.

Communicational Cost: In our scheme, a tag is one element of $\mathbb{G}_1$, the challenge size is bounded of $3|Z_q|$, the proof size is one element of $\mathbb{G}_1$, one element of $\mathbb{G}_2$ and one element of $|Z_q|$. The total communicational cost of our scheme is very low. We also compare our scheme with another three similar schemes in Table 2.

From Table 2, we can find that the size of proof in our scheme is a little longer than others, but the gap is very small and keeps constant. However, our scheme has a great advantage in terms of challenge size, which still increases with the incrementing of U and c. Thus, our scheme has better communication performance.

B. EXPERIMENT RESULTS

We implemented a prototype of our scheme with PBC library [41] which is based on the library of GMP [42]. Our experiments set the workgroup with 100 users and the size of the data shared in the group is 2M. The experiments are executed in ubuntukylin-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 Win10 operation system with Core i5 CPU and 8G Ram. We choose the typical 'Type A' elliptic curve supplied

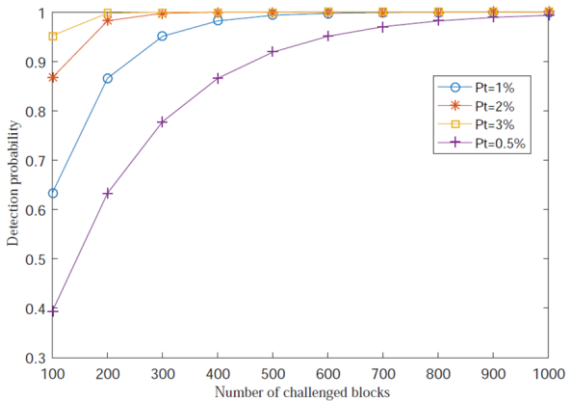

FIGURE 2. Probability of misbehavior detection.

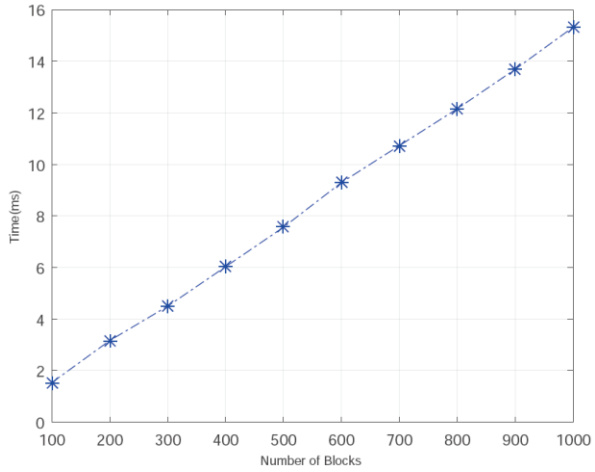

FIGURE 3. Cost of tag generation.

by PBC in our experiments. In order to accurately show the advantage of our scheme, we implement CL-PGSDP and CLCA schemes simultaneously.

First, we execute experiments to evaluate the performance of tag generation in our scheme. In these experiments, we generate 100 to 1,000 tags for different data blocks. The experimental results are shown in Figure.3. Looked from the overall, the cost of tag generation is linear with the number of data blocks. To generate 1,000 tags needs only about 15.2 seconds which is efficient for practical application. Furthermore, if the computation of *TagGen* is done offline, the cost will decrease greatly. Besides, each tag is generated for only one time, so that it brings little impact on the entire performance of the scheme.

The second experiment is to evaluate the performance of 'Challenge' phase. In this experiment, the count of group users is 100, and the count of challenged blocks changes from 100 to 500. The experiment data is shown in Figure 4. From the Figure 4, we can find that the cost of 'CL-PGSDP' increases linearly with the increment of challenged blocks and much greater than that of 'CLCA' and our scheme. The challenge cost of 'CLCA' scheme is almost invariable, because its cost is related to the count of users not

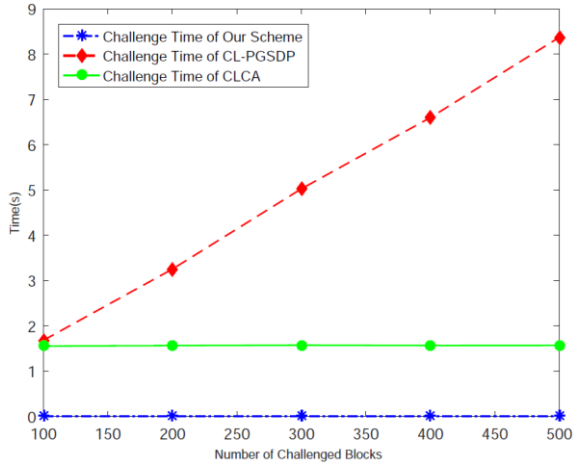

FIGURE 4. Cost of challenge.

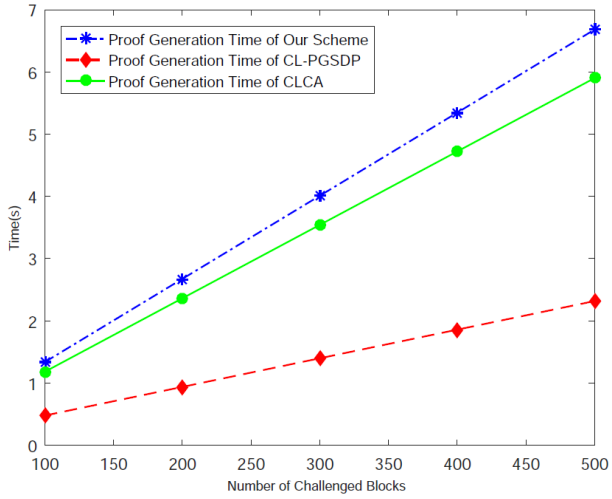

FIGURE 5. Cost of proof generation.

the challenged blocks. Our scheme has negligible cost and is much more efficient than the others.

Figure 5 demonstrates the computational cost of the 'proof generation' phase. We can see that our scheme needs more computation cost than CL-PGSDP and CLCA in this phase. According to the Figure 2, if $P_t = 1\%$, we can use about 400 blocks to achieve 98% probability of misbehavior detection. Under this condition, our scheme only needs 4 seconds more than CL-PGSDP scheme. Moreover, the work of proof generation is taken by CSP. Since CSP has great computational ability, the gap of computation cost in this phase has negligible impact on the entire efficiency of the scheme.

The cost of proof audit is presented in Figure 6. We can see that all three schemes consume linear cost with number of challenged blocks in verification phase. CL-PGSDP scheme costs greater overhead than CLCA scheme and our scheme. CLCA scheme has similar cost to ours, but still higher than our scheme. In summary, our scheme is the most efficient one in this phase.

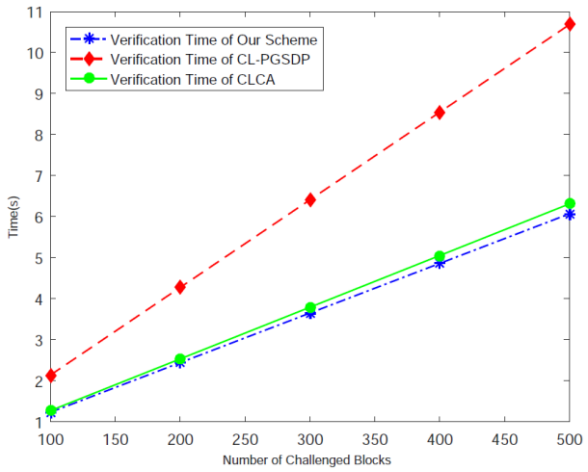

FIGURE 6. Cost of proof verification.

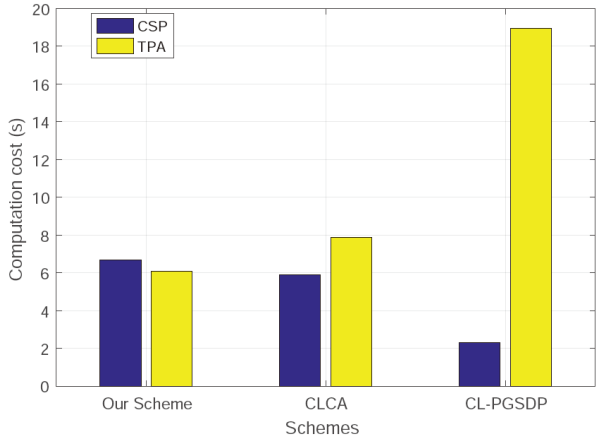

FIGURE 7. Computation cost of TPA and CSP.

At last, we make experiments to summary the computation cost of CSP and TPA in the three schemes. The number of challenged block is set to 500. The results are shown in Figure 7.

Observed from the Figure 7, the TPA in CL-PGSDP assumes more computiation cost than that of in our scheme and CLCA scheme. Specifically, our scheme assigns the lightest workload to TPA. Furthermore, in CL-PGSDP the computation cost of CSP is much lower than the computation cost of TPA. However, in our schme, the suituation is opposite. It is well known that CSP has greate computation ability but TPA is usually a normal workstation or personal computer. Transferring more job from TPA to CSP is a reasonable way to improve the efficiency of PDP scheme. Thus, our scheme realizes a better mechnism than the others.

Overall, compared with recent researches, our scheme is efficient especially for TPA.

VII. CONCLUSION

In this paper, we present a public identity-based PDP protocol for secure data storage, which supports identity privacy protection of multiple users. With our scheme, TPA can check the integrity of group shared data rightly but cannot know who uploaded the challenged data. We give the security model for our scheme, and prove its security with features of completeness, soundness and identity privacy preserving. Experimental result demonstrates that our proposal is efficient.

REFERENCES

[1] M. Ali, R. Dhamotharan, E. Khan, S. U. Khan, A. V. Vasilakos, K. Li, and A. Y. Zomaya, "SeDaSC: Secure data sharing in clouds," *IEEE Syst. J.*, vol. 11, no. 2, pp. 395–404, Jun. 2017.

[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 Trans. Dependable Secure Comput.*, early access, Jan. 3, 2020, doi: 10.1109/TDSC.2020.2963978.

[3] G. Chunpeng, Z. Liu, J. Xia, and F. Liming, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Trans. Dependable Secure Comput.*, early access, Feb. 14, 2019, doi: 10.1109/TDSC.2019.2899300.

[4] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proc. Conf. Hot Topics Cloud Comput.*, San Diego, CA, USA, 2009, pp. 14–19.

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

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

[7] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, Alexandria, VA, USA, 2007, pp. 598–609.

[8] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Melbourne, VIC, Australia, 2008, pp. 90–107.

[9] A. Juels and B. S. Kaliski, "PORs: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 584–597.

[10] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netowrks (SecureComm)*, 2008, pp. 1–10.

[11] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS)*, 2009, pp. 213–222.

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

[13] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2402–2415, Oct. 2017.

[14] 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 Trans. Comput.*, vol. 64, no. 9, pp. 2609–2622, Sep. 2015.

[15] H. Wang, "Identity-based distributed provable data possession in multi-cloud storage," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 328–340, Mar. 2015.

[16] J. Li, H. Yan, and Y. Zhang, "Efficient identity-based provable multi-copy data possession in multi-cloud storage," *IEEE Trans. Cloud Comput.*, early access, Jul. 16, 2019, doi: 10.1109/TCC.2019.2929045.

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

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

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

[20] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min, "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 767–778, Apr. 2017.

[21] 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 Gener. Comput. Syst.*, vol. 76, pp. 136–145, Nov. 2017.
[22] B. Wang, B. Li, and H. Li, "Knox: Privacy-preserving auditing for shared data with large groups in the cloud," in *Proc. 10th Int. Conf. Appl. Cryptogr. Netw. Secur. (ACNS)*, 2012, pp. 507–525.
[23] B. Wang, H. Li, and M. Li, "Privacy-preserving public auditing for shared cloud data supporting group dynamics," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 1946–1950.
[24] L. Chen, J. Li, and Y. Zhang, "Anonymous certificate-based broadcast encryption with personalized messages," *IEEE Trans. Broadcast.*, vol. 66, no. 4, pp. 867–881, Dec. 2020, doi: 10.1109/TBC.2020.2984974.
[25] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure multi-owner data sharing for dynamic groups in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1182–1191, Jun. 2013.
[26] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," *IEEE Trans. Services Comput.*, vol. 8, no. 1, pp. 92–106, Jan. 2015.
[27] 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," *J. Syst. Softw.*, vol. 113, pp. 130–139, Mar. 2016.
[28] G. Wu, Y. Mu, W. Susilo, and F. Guo, "Privacy-preserving cloud auditing with multiple uploaders," in *Proc. Int. Conf. Inf. Secur. Pract. Exper. (ISPEC)*, 2016, pp. 224–237.
[29] S. K. Nayak and S. Tripathy, "SEPDP: Secure and efficient privacy preserving provable data possession in cloud storage," *IEEE Trans. Services Comput.*, early access, Mar. 29, 2018, doi: 10.1109/TSC.2018.2820713.
[30] J. Yu and R. Hao, "Comments on 'SEPDP: Secure and efficient privacy preserving provable data possession in cloud storage,'" *IEEE Trans. Services Comput.*, early access, Apr. 23, 2019, doi: 10.1109/TSC.2019.2912379.
[31] G. C. Mara, U. Rathod, R. R. G. Shreyas, S. Raghavendra, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "CRUPA: Collusion resistant user revocable public auditing of shared data in cloud," *J. Cloud Comput.*, vol. 9, no. 1, pp. 1–18, Dec. 2020.
[32] X. Lu, Z. Pan, and H. Xian, "An efficient and secure data sharing scheme for mobile devices in cloud computing," *J. Cloud Comput.*, vol. 9, no. 1, pp. 1–13, Dec. 2020.
[33] 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 *Proc. 8th Int. Conf. Netw. Syst. Secur. (NSS)*, 2014, pp. 28–40.
[34] J. Li, H. Yan, and Y. Zhang, "Certificateless public integrity checking of group shared data on cloud storage," *IEEE Trans. Services Comput.*, vol. 14, no. 1, pp. 71–81, Jan./Feb. 2021, doi: 10.1109/TSC.2018.2789893.
[35] H. Yang, S. Jiang, W. Shen, and Z. Lei, "Certificateless provable group shared data possession with comprehensive privacy preservation for cloud storage," *Future Internet*, vol. 10, no. 6, p. 49, Jun. 2018.
[36] G. Wu, Y. Mu, W. Susilo, F. Guo, and F. Zhang, "Privacy-preserving certificateless cloud auditing with multiple users," *Wireless Pers. Commun.*, vol. 106, no. 3, pp. 1161–1182, Jun. 2019.
[37] J. Li, L. Chen, Y. Lu, and Y. Zhang, "Anonymous certificate-based broadcast encryption with constant decryption cost," *Inf. Sci.*, vols. 454–455, pp. 110–127, Jul. 2018.
[38] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, 2001, pp. 213–229.
[39] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Taipei, Taiwan, 2003, pp. 452–473.
[40] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, Sep. 2004.
[41] *The Pairing-based Cryptography Library (PBC)*. Accessed: Feb. 15, 2020. [Online]. Available: https://crpto.stanford.edu/pbc/download.html
[42] *The GNU Multiple Precision Arithmetic Library (GMP)*. Accessed: Feb. 15, 2020. [Online]. Available: http://gmplib.org/

HAO YAN received the B.S. and M.S. degrees in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science and technology from Hohai University, Nanjing, in 2019. He is currently an Associate Professor with the School of Network Security, Jinling Institute of Technology, Nanjing. His research interests include cryptography and information security, cloud computing, networks security, and trusted computing.

WENMING GUI received the B.S., M.S., and Ph.D. degrees in computer science and technology in 1996, 2002, and 2013, respectively. He is currently an Associate Professor with the Jinling Institute of Technology, Nanjing, China. His research interests include networks security, machine learning, and music artificial intelligence.