IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 14, 2020, accepted July 12, 2020, date of publication July 15, 2020, date of current version July 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009539

Privacy-Preserving Cloud Auditing for Multiple Users Scheme With Authorization and Traceability

XIAODONG YANG[1], (Member, IEEE), MEIDING WANG[1], TING LI[1], RUI LIU[1], AND CAIFEN WANG[1,2]

[1]College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
[2]College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China

Corresponding author: Caifen Wang (wangcfen@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61662069 and Grant 61562077, in part by the China Postdoctoral Science Foundation under Grant 2017M610817, in part by the Science and Technology Project of Lanzhou City under Grant 2013-4-22, and in part by the Foundation of Northwest Normal University under Grant NWNU-LKQN-14-7.

ABSTRACT With the widespread application of cloud storage, users could obtain many conveniences such as low-price data remote storage and flexible data sharing. Considering cloud service provider (CSP) is not full-trusted, lots of cloud auditing schemes are proposed to ensure the shared data security and integrity. However, existing cloud auditing schemes have some security risks, such as user identity disclosure, denial of service attack and single-manager abuse of power. To solve the above issues, we use certificateless signature technology to construct a privacy-preserving cloud auditing scheme for multiple users with authorization and traceability in this paper. Unlike the traditional schemes, our scheme realizes user identity anonymity without group signature and ring signature techniques, which guarantees the tag is compact. Meanwhile, our scheme supports that at least d managers could trace the identity of malicious user collaboratively, which avoids the abuse of single-manager power and provides non-frameability. Furthermore, we introduce an identity authentication process between the third-party auditor (TPA) and the CSP to prevent the denial of service attack. That is, our scheme could solve the problem that anyone can challenge the CSP for the proofs, which averts network congestion and waste of cloud resources. In terms of function, the proposed scheme also supports efficient user revocation from a group. Certificateless cryptography ensures that our scheme does not involve certificate management burden and the key escrow problem. The security analysis shows that our scheme is provably secure against two types of adversaries in the environment of certificateless cryptography. The performance analysis demonstrates that our scheme is efficient.

INDEX TERMS Authorization, certificateless cryptography, cloud auditing, privacy-preserving, revocation, traceability.

I. INTRODUCTION

With the rapid development of computer technology, people need to process and store a lot of data every day. To reduce the cost of data management and infrastructure maintenance, users choose to store their data files in the cloud. Unfortunately, the security and the integrity of data in the cloud is being challenged. On the one hand, software failures and physical device damage [1] may cause shared data loss. On the other hand, malicious CSP may modify or delete data in the cloud storage for their benefit [2], [3]. Since users lose the direct control of data on cloud storage, they cannot determine whether their data are still complete. In particular, it is not feasible for users to download all the data files to check them because of the expensive communication costs. Therefore, designing an effective way that could audit the integrity of data in the cloud without downloading them is very important.

To confirm the integrity of cloud data, researchers proposed a large number of cloud auditing schemes. In these schemes, the TPA is allowed to check the integrity of shared data on behalf of users which reduces the heavy burden

The associate editor coordinating the review of this manuscript and approving it for publication was Gautam Srivastava.

of users. However, most of the cloud auditing schemes are based on traditional public key cryptography (TPKC). Despite the extensive application background of TPKC, there are still some problems. For example, TPKC requires certificate authorizer (CA) to generate certificates that bind users' identities and the associated public keys. As the number of users increases, the certificate management becomes more difficult, which causes the existing cloud auditing schemes not to apply for groups with multiple users. To avoid the shortcoming, some researchers choose identity-based cryptography (ID-PKC) to be the basis in their schemes. Unfortunately, ID-PKC requires the private key generator (PKG) to distribute corresponding private key to every user. Once PKG becomes untrustworthy, the scheme is no longer safe. In this case, certificateless cryptography is a choice since it does not involve certificates management nor depend on the other fully-trusted entity. There is a third-party entity named the key generation center (KGC) to be responsible for generating system master key, public parameters, and partial keys of users. Thus, constructing cloud auditing schemes based on certificateless cryptography is a reliable way to check the security and the integrity of shared data.

In some existing certificateless cloud auditing schemes, there is usually only one user to upload tags and request the TPA to audit data. To meet the uploading requirement of multiple users, some schemes [4], [5] support that verifiers could batch check the correctness of the multiple tags. However, there are multiple users to send request to the TPA in a group that would bring the new security flaw. For instance, the challenge is generated by the uploader's public key, so the TPA could find the user who uploaded the file by the challenge. However, the identity of the user needs to remain anonymous in plenty of situations, such as an electronic voting system. For the problem, researchers presented some privacy-preserving schemes for group users to guarantee that their identities are anonymous to anyone. On the one hand, these schemes guarantee the identity privacy of users; on the other hand, it is difficult to trace the identities of the misbehaved users. A later development is cloud auditing scheme based on group signature [6], which realizes the user privacy-preserving meanwhile reveals the user's identity after a dispute. Unfortunately, the identity tracing of the scheme [6] relies on a single group manager. The single group manager has extremely high permissions, which may lead to the innocent users being framed and the malicious users being sheltered.

Meanwhile, a requisite identity authentication process is often lacking between the TPA and the CSP in many existing cloud auditing schemes. It causes that any entity could frequently send request to the CSP for getting the auditing proof by utilizing the TPA. In this situation, the malicious or pretended users may launch denial of service attacks in the cloud, which brings network congestion or waste of resources. Hence, it is important to solve the unauthorized problem in the cloud auditing for shared data.

Moreover, considering the group is dynamic, every group user needs to be allowed to leave at any moment. Therefore, the user revocation from the group is a significant, yet timely issue. To be specific, once a user revoked from the group, he/she must lose the right to access the data in the cloud and declare that all group information of the revoked user is invalid. Besides, the tags of all shared data that are generated by the revoked user also need to be updated. The traditional method of tag updating is that the trusted user downloads these files, re-signs new tags for these files, and uploads them to the CSP. But the method increases the computation and communication costs of group users. Therefore, CSP is more suitable for finishing the task of user revocation than ordinary users.

The above problems can be summarized into three aspects: firstly, most existing cloud auditing schemes cannot simultaneously satisfy user's identity anonymity and traceability. Some schemes rely on single-manager who may frame the innocent users and hide the misbehaving users in the process of tracing. Secondly, lacking an effective identity authorization process between the TPA and the CSP could lead to problems such as network congestion and the waste of network resources. Thirdly, the heavy computation and communication costs for the revocation of group users affect the quality of the cloud storage service. Therefore, we consider it is significant to design a cloud auditing scheme that can support privacy-preserving of users, identity trace of the malicious user, identity authorization and efficient user revocation under the certificateless cryptography at the same time.

A. RELATED WORKS

For the shared data stored in the cloud, users cannot download all of them to check their integrity. Thus, the schemes of traditional cryptography could not be directly applied to the integrity checking of data in the cloud [7]. To solve the problem, Shacham and Waters [8] proposed the concept of public auditing for the first time. Then they presented the first cloud auditing scheme, which could check whether the data in the cloud are complete. For better performance and security, a batch of cloud auditing schemes based on TPKC [9]–[14] are proposed. In these schemes, CA is responsible for certificates distribution and certificates storage. These complex operations would bring huge costs as the number of users increases. In 1984, Shamir [15] proposed the concept of ID-PKC, which solves the problem of certificates management of TPKC. Wang *et al.* [16] proposed the first public cloud auditing scheme based on ID-PKC and showed it is provable security. Li *et al.* [17] proposed a multi-copy on distributed cloud servers auditing scheme based on ID-PKC. Later, Li *et al.* [18] proposed a cloud auditing scheme based on ID-PKC, which realizes the data privacy against the TPA. However, in all cloud auditing schemes based on ID-PKC [19], [20], the user's private key is provided by KGC. KGC has an ability to generate the tag of any user, which may reveal the user's identity information. Therefore,

these schemes have the problem of key escrow. Al-Riyami and Paterson [21] proposed the concept of certificateless cryptography. It solves the inherent flaws of both cryptographies. Wang *et al*. [12] presented the first certificateless public cloud auditing scheme. However, scheme [12] cannot resist the first type of adversary. That is, the adversary could replace the user's public key. Moreover, when malicious TPA appears, the scheme does not support user privacy-preserving.

For the study of privacy-preserving, researchers usually divide it into two aspects: data privacy-preserving [22], [23] and user privacy-preserving [14], [24]–[26]. The main technologies of user privacy-preserving are blind signature [27], ring signature and group signature. In certificateless settings, cloud auditing schemes realize user privacy-preserving by using certificateless ring signatures [28], [29] or zero-knowledge proof [10] technologies, which have the non-compact tags, resulting in most of the storage space is allocated to tags rather than data. Wu *et al.* [30] combined the private key extraction algorithm of the Water's IBE scheme with the tag generation algorithm of the scheme [31] and presented a cloud auditing scheme with privacy-preserving to solve the above problems. The scheme achieves unconditional privacy-preserving, but cannot support group users' identity trace and the revocation of group users. Li *et al.* [32] presented a privacy-preserving cloud auditing scheme with attribute-based encryption, which also cannot support the revocation of group users.

To deal with the revocation problem of group users, Yuan and Yu [33] presented a polynomial based authentication tag scheme for shared data. Yu *et al.* [34] constructed a cloud auditing scheme without pairing that allows group users to join and revoke. Unfortunately, both schemes have the problem of key escrow. They are inefficient. To cut down the computation overhead, proxy re-signature technology [35] becomes a good choice. Wang *et al.* [6] designed a cloud auditing scheme for data that utilizes the proxy re-signature. Under the cooperation of the CSP and the revoked user, the tags of the revoked user are translated into the tags of the trusted user. However, in order to generate resignation keys of new tags, the CSP in scheme [6] must obtain the resignation key of every group user in advance, which can cause security issues. In 2018, Li *et al.* [36] designed a certificateless public auditing scheme for cloud data, which can support efficient user revocation. But it does not realize the functions of identity traceability and privacy-preserving. Moreover, to prevent disguised TPA from sending unauthorized data verification challenges to CSP, Liu *et al.* [37] presented a cloud auditing scheme that introduces the authorization process between the TPA and the CSP. The authorization scheme utilizes Merkle Hash Tree and BLS signatures to achieve fine-grained update requirements, but it cannot be applied to groups.

B. CONTRIBUTIONS

In this paper, a privacy-preserving cloud auditing scheme for multiple users with authorization and traceability is proposed. Our contributions are summarized as follows.

(1) We propose a cloud auditing scheme with group users based on certificateless cryptography to avoid the certificate management issues of TPKC and key escrow issues of ID-PKC.

(2) Our scheme achieves the user's identity privacy-preserving against the TPA. Since the proposed scheme does not use group signature and ring signature technologies, the tag is compact. This could ensure efficient storage of shared data.

(3) Our scheme has an ability to reveal the real identity of the malicious user while keeping the group user's identity anonymous. Besides, based on the Lagrange interpolation, the proposed scheme allows at least d group managers could trace the identity of the misbehaved user in the group. The method provides non-frameability, which can guarantee the fairness of tracing and avoid framing the innocent user and shielding the malicious user.

(4) We introduce an identity authorization process to check the validity of the challenge message from the TPA. That is, only the TPA approved by group users could obtain the proof of the CSP, which protects the CSP from malicious harassment.

(5) The proposed scheme supports efficient group user revocation. In the process of revocation, our scheme does not require complex operations such as downloading, re-signing, and re-uploading, which reduces the computation and communication burden.

(6) We prove that our scheme is safe against two types of adversaries in certificateless cryptography. The analysis and experiments demonstrate that the proposed scheme is more efficient than other similar schemes.

II. PRELIMINARIES

We introduce the related knowledge used in our scheme, which includes bilinear pairing, hardness assumptions, the system model, the definition of our scheme and the security model.

A. BILINEAR PAIRING

Let G_1 and G_2 be two cyclic groups of prime order p. The bilinear map $e : G_1 \times G_1 \rightarrow G_2$ satisfies the following conditions:

(1) Bilinearity: For any $g_5, g_6 \in G_1, a_1, b_1 \in Z_p$, there is $e(a_1 g_5, b_1 g_6) = e(g_5, g_6)^{a_1 b_1}$.

(2) Non-degeneracy: For $g_5, g_6 \in G_1$, there is $e(g_5, g_6) \neq 1$.

(3) Computability: For any $g_5, g_6 \in G_1$, $e(g_5, g_6)$ can be calculated.

B. HARDNESS ASSUMPTIONS

Double-BDH problem: Given the pairing group $PG = (G_1, G_2, e, g, p)$, where the p is prime order of the groups and g is a generator of G_1. Choose a tuple $\left(g, g^a, g^b, g^c\right)$, where $a, b, c \in Z_p$. Compute $e(g, g)^{\frac{ac}{b}}$.

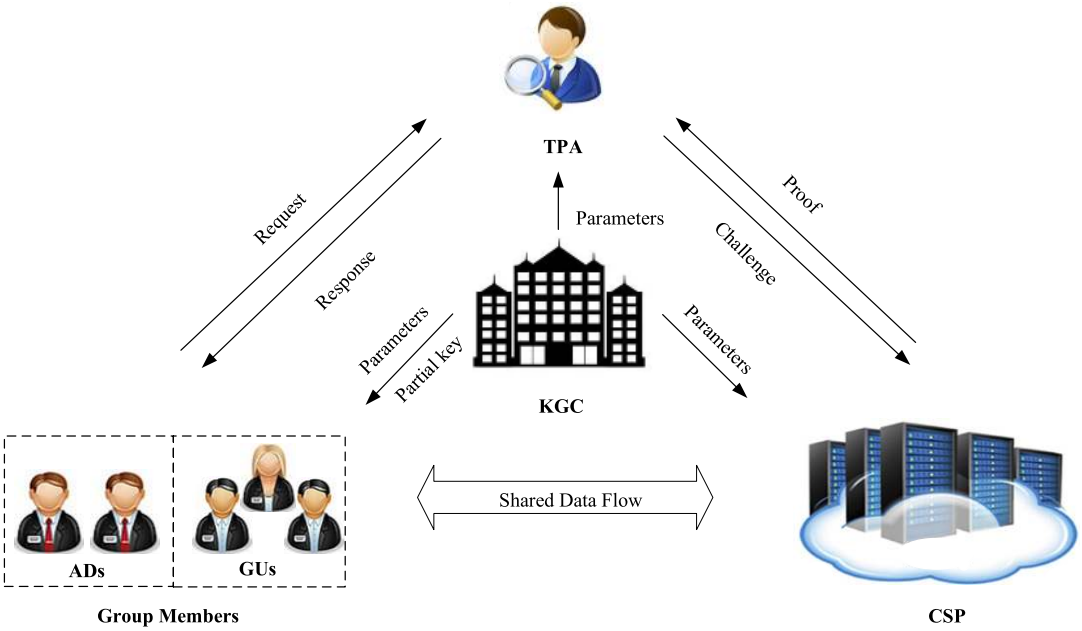

FIGURE 1. System model.

Definition 1 (Double-BDH Hypothesis): The probability of polynomial time algorithm B in solving the double-BDH problem is defined as:

$$Adv_{double-BDH}(B) = \Pr\left[\left\{e(g,g)^{\frac{ac}{b}}\right\} \longleftarrow B\left(PG, g, g^a, g^b, g^c\right)\right].$$

If $Adv_{double-BDH}(B)$ is negligible, it is difficult to solve the double-BDH problem.

vBDH Problem: Given the pairing group $PG = (G_1, G_2, e, g, p)$, where the p is prime order of the groups and g is a generator of G_1. Choose a tuple $\left(g, g^a, g^b, g^{ac}\right)$, where $a, b, c \in Z_p$. Compute $e(g,g)^{bc}$.

Definition 2 (vBDH Hypothesis): The probability of polynomial time algorithm B in solving the vBDH problem is defined as:

$$Adv_{vBDH}(B) = \Pr\left[\left\{e(g,g)^{bc}\right\} \longleftarrow B\left(PG, g, g^a, g^b, g^{ac}\right)\right].$$

If $Adv_{vBDH}(B)$ is negligible, it is difficult to solve the vBDH problem.

C. SYSTEM MODEL

In this subsection, we explain the system model of our scheme, which is shown in Figure 1. The system model in the paper mainly involves five entities: KGC, the group users (GUs), the group managers (ADs), CSP and TPA. The specific interaction processes are as follows. Firstly, GUs generate their private keys and public keys with the partial keys distributed by KGC. Secondly, GUs generate tags for shared data with their private keys, and store shared data and tags in the cloud. Thirdly, GUs send their requests to the TPA for auditing the integrity of shared data. After receiving the requests, the TPA sends the challenge to the CSP for getting the auditing proof. The CSP authenticates the identity of the TPA. If it is valid, the CSP generates the proof and sends it to the TPA. Otherwise, the CSP refuses to accept the challenge. Finally, the TPA checks the effectiveness of the proof from the CSP and gives a result about shared data integrity to GUs.

(1) **KGC**: It is not a fully-trusted third-party entity that is responsible for generating system master key, public parameters, partial keys of group members.

(2) **GUs**: They are group users of the system who have an ability to access, upload, modify, and delete the shared data in the cloud.

(3) **ADs**: They are part of the group members, responsible for the revocation of GUs. A certain number of ADs could trace the real identity of the malicious user.

(4) **CSP**: It is a semi-trusted third-party entity that is responsible for storing and sharing data.

(5) **TPA**: It is the auditor who sends the challenge to the CSP. The CSP generates the response and sends it to the TPA. If the response passes the verification, the data is complete. Otherwise, the data is damaged or tampered.

D. DEFINITION OF OUR SCHEME

The proposed scheme in this paper is composed of nine algorithms: Setup, Extract-Partial-Key, Set-Secret-Value, Set-Private-Key, Set-Public-Key, Tag-Generation, Challenge-Generation, Proof-Generation and Proof-Verification. The detailed descriptions of these algorithms are as follows:

Setup $(o) \rightarrow (\partial, param)$: Input the security parameter o, KGC outputs the system master key ∂ and the public parameter $param$.

Extract-Partial-Key $(\partial, param, ID_i) \rightarrow (D_i)$: Input the system master key ∂, the public parameter *param* and the identity ID_i, KGC outputs the partial key D_i.

Set-Secret-Value $(param, ID_i) \rightarrow (\beta_i)$: Input the public parameter *param* and the identity ID_i, u_i returns the secret value β_i.

Set-Private-Key $(param, D_i, \beta_i) \rightarrow (sk_i)$: Input the public parameter *param*, the partial key D_i and the secret value β_i, u_i returns the private key sk_i.

Set-Public-Key $(param, D_i, \beta_i) \rightarrow (pk_i)$: Input the public parameter *param*, the partial key D_i and the secret value β_i, u_i returns the public key pk_i.

Tag-Generation $(param, sk_i, pk_i, m, j) \rightarrow (\sigma_j, P_i)$: Input the public parameter *param*, the private key sk_i, the public key pk_i, the data m and the index j, u_i generates the tag σ_j and P_i.

Challenge-Generation $(param, PK, ID, j_{\max}) \rightarrow (\Psi)$: Input the public parameter *param*, the public keys PK of GUs, the identity information ID of GUs and the maximum index $j_{\max}$, TPA generates the challenge Ψ.

Proof-Generation $(param, \Psi, \overline{M}, \Lambda) \rightarrow (\Omega)$: Input the public parameter *param*, the challenge Ψ, the data $\overline{M}$ and the tags Λ, CSP generates the proof Ω.

Proof-Verification $(param, \Omega) \rightarrow$ (“1”, “0”): Input the public parameter *param* and the proof Ω, TPA outputs “1” or “0”, where 1 represents the verification succeed, 0 represents the verification failed.

E. SECURITY MODEL

Our scheme focuses on two security issues, unforgeability and anonymity. Unforgeability means that no one can forge the legitimate tags of the group user u_i. We introduce two adversaries to demonstrate the unforgeability in our security model. $\mathcal{A}_{\text{I}}$ is the adversary who can replace the public key of any group user, although he/she does not know the system master key ∂. $\mathcal{A}_{\text{II}}$ is the adversary who has the system master key ∂, but cannot replace the public key of any group user. Anonymity refers to that adversary $\mathcal{A}_{\text{III}}$ can't distinguish which user uploads the files in the process of data auditing. This ensures the privacy security of group users. The security model is implemented by playing games between adversaries and challengers.

Game I: Adversary $\mathcal{A}_{\text{I}}$ and challenger $\mathcal{C}$ simulate this game.

Setup: $\mathcal{C}$ runs the System-Parameter algorithm, saves system master key ∂ and sends public parameter *param* to $\mathcal{A}_{\text{I}}$.

Queries: $\mathcal{A}_{\text{I}}$ makes polynomial times queries to $\mathcal{C}$, and $\mathcal{C}$ answers the queries of $\mathcal{A}_{\text{I}}$ as follows.

- Hash-Query: $\mathcal{A}_{\text{I}}$ asks $\mathcal{C}$ for hash values. $\mathcal{C}$ sends the result to $\mathcal{A}_{\text{I}}$.
- Partial-Key-Query: $\mathcal{A}_{\text{I}}$ submits identity ID to $\mathcal{C}$ and queries the partial key of ID. $\mathcal{C}$ runs the Extract-Partial-Key algorithm to generate the partial key of ID, and sends it to $\mathcal{A}_{\text{I}}$.
- Secret-Value-Query: $\mathcal{A}_{\text{I}}$ submits identity ID to $\mathcal{C}$, and inquires the secret value of ID. $\mathcal{C}$ generates the secret value of ID by running the Set-Secret-Value algorithm. $\mathcal{C}$ sends the secret value to $\mathcal{A}_{\text{I}}$.
- Public-Key-Query: $\mathcal{A}_{\text{I}}$ submits identity ID to $\mathcal{C}$, and inquires the public key of ID. $\mathcal{C}$ generates the public key of ID by running the Set-Public-Key algorithm. $\mathcal{C}$ sends the public key to $\mathcal{A}_{\text{I}}$.
- Public-Key-Replacement: $\mathcal{A}_{\text{I}}$ selects values to replace the public key of ID.
- Tag-Query: $\mathcal{A}_{\text{I}}$ submits identity ID and data m to $\mathcal{C}$, and inquires the tag of m that signed by ID. $\mathcal{C}$ generates the tag σ of ID by running the Tag-Generation algorithm. $\mathcal{C}$ sends σ to $\mathcal{A}_{\text{I}}$.

Forgery: $\mathcal{A}_{\text{I}}$ outputs the tag σ' of the identity ID' to data m' under the public key $pk_{ID'}$.

$\mathcal{A}_{\text{I}}$ wins the game if the following conditions are satisfied.

(1) $\mathcal{A}_{\text{I}}$ never queries about the partial key of the identity ID'.

(2) $\mathcal{A}_{\text{I}}$ never queries about the private key of the identity ID'.

(3) $\mathcal{A}_{\text{I}}$ never queries about the tag of data m' under the public key $pk_{ID'}$ of ID'.

(4) The tag σ' generated by $\mathcal{A}_{\text{I}}$ is valid.

Game II: Adversary $\mathcal{A}_{\text{II}}$ and challenger $\mathcal{C}$ simulate this game.

Setup: $\mathcal{C}$ runs the System-Parameter algorithm, sends system master key ∂ and public parameter *param* to $\mathcal{A}_{\text{II}}$.

Queries: $\mathcal{A}_{\text{II}}$ makes polynomial times queries to $\mathcal{C}$, and $\mathcal{C}$ answers the queries of $\mathcal{A}_{\text{II}}$ as follows.

- Hash-Query: $\mathcal{A}_{\text{II}}$ asks $\mathcal{C}$ for hash values. $\mathcal{C}$ sends the result to $\mathcal{A}_{\text{II}}$.
- Secret-Value-Query: $\mathcal{A}_{\text{II}}$ submits identity ID to $\mathcal{C}$, and inquires the secret value of ID. $\mathcal{C}$ generates the secret value of ID by running the Set-Secret-Value algorithm. $\mathcal{C}$ sends the of ID by running the Set-Secret-Value algorithm. $\mathcal{C}$ sends the secret value to $\mathcal{A}_{\text{II}}$.
- Public-Key-Query: $\mathcal{A}_{\text{II}}$ submits identity ID to $\mathcal{C}$, and inquires the public key of ID. $\mathcal{C}$ generates the public key of ID by running the Set-Public-Key algorithm. $\mathcal{C}$ sends the public key to $\mathcal{A}_{\text{II}}$.
- Tag-Query: $\mathcal{A}_{\text{II}}$ submits identity ID and data m_j to $\mathcal{C}$, then inquires the tag of m that signed by ID. $\mathcal{C}$ generates the tag σ of ID by running the Tag-Generation algorithm. $\mathcal{C}$ sends σ to $\mathcal{A}_{\text{II}}$.

Forgery: $\mathcal{A}_{\text{II}}$ outputs the tag σ' of the identity ID' to data m'.

$\mathcal{A}_{\text{II}}$ wins the game if the following conditions are satisfied.

(1) $\mathcal{A}_{\text{II}}$ never queries about the secret value of the identity ID'.

(2) $\mathcal{A}_{\text{II}}$ never queries about the tag of data m' with ID'.

(3) The tag σ' generated by $\mathcal{A}_{\text{II}}$ is valid

Game III: The game proves the anonymity of the scheme between adversary $\mathcal{A}_{\text{III}}$ and challenger $\mathcal{C}$.

Setup: $\mathcal{C}$ generates the system master key ∂ and public parameters *param* by running the Setup algorithm. $\mathcal{C}$ selects n users, performs the Extract-Partial-Key algorithm and the Set-Secret-Value algorithm to generate partial keys and secret

values for them. $\mathcal{C}$ generates identities of users and public keys, then sends public parameters, $ID = (ID_1, \ldots, ID_n)$ and $PK = \left(pk_{ID_1}, \ldots, pk_{ID_n}\right)$ to $\mathcal{A}_{|||}$.

Challenge: $\mathcal{A}_{|||}$ selects any data m, runs the Challenge algorithm. Then $\mathcal{A}_{|||}$ sends the challenge Ψ and data m to $\mathcal{C}$.

Response: $\mathcal{A}_{|||}$ randomly selects the user's identity ID, generates the tag with the private key of ID and runs the Proof-Generation algorithm to generate Ω, then sends the response Ω to $\mathcal{C}$.

Guess: $\mathcal{A}_{|||}$ checks Ω. If it is valid, $\mathcal{A}_{|||}$ outputs the identity ID^* who uploads m.

$\mathcal{A}_{|||}$ wins the game if $ID = ID^*$.

If the advantage of our scheme in breaking the anonymity is 0, our scheme satisfies theoretically anonymous.

III. THE PROPOSED SCHEME

A. CONSTRUCTION OF OUR SCHEME

In our scheme, there are Z group managers and S group users. We define ID_i as the identity of the group user u_i for $1 \leq i \leq S$. We set $m \in Z_p$ to be the shared data. The construction is described as follows.

(1) **Setup**: KGC generates the system master key and the public parameter as follows.

• KGC generates two multiplicative cyclic groups G_1 and G_2 of order p, which satisfy the bilinear pairing $e : G_1 \times G_1 \to G_2$. g is the generator of G_1.

• KGC chooses two hash functions $H_1, H_2 : \{0, 1\}^* \to G_1$.

• KGC randomly chooses $\partial \in Z_p$ as the system master key, calculates $g_1 = g^{\partial}$.

• KGC randomly chooses $v \in G_1, ask \in Z_p, apk \in Z_p, csk \in Z_p, cpk \in Z_p$. (ask, apk) and (csk, cpk) are the key pairs used to authorize.

• KGC randomly selects $k \in Z_p$, computes $g_2 = g^k$, sets up a $d-1$ degree polynomial $f(x) = c_0 + c_1 x + \cdots + c_{d-1} x^{d-1}$ with $c_0 = k, c_1, \cdots c_{d-1} \in Z_p$, computes $k_l = f(l)$ $(l = 1, 2, \cdots, Z$ and $2d - 1 \geq Z)$, divides k into Z pieces k_l, sends them to ADs.

• KGC keeps the system master key secretly and sets the public parameter $param = \{G_1, G_2, e, g, g_1, g_2, v, H_1, H_2\}$.

(2) **Extract-Partial-Key**: On input the system master key ∂ and the identity ID_i, KGC follows the steps to generate the partial key of u_i.

• KGC randomly chooses $\omega_i \in Z_p$.

• KGC calculates the partial key $D_i = (d_{i1}, d_{i2})$, where $d_{i1} = H_1 (ID_i)^{\partial}$, $d_{i2} = \omega_i$.

• KGC returns D_i to u_i.

(3) **Set-Secret-Value**: u_i randomly chooses $\beta_i \in Z_p$ as the secret value.

(4) **Set-Private-Key**: On input the partial key D_i and the secret value β_i, u_i follows the steps to generate the private key.

• u_i calculates $sk_i = (sk_{i1}, sk_{i2}) = \left(\left(H_1 (ID_i)^{\partial}\right)^{\omega_i \cdot \beta_i}, \beta_i\right)$ as his/her private key.

(5) **Set-Public-Key**: On input the partial key D_i and the secret value β_i, u_i follows the steps to generate the public key.

• u_i calculates $pk_i = (pk_{i1}, pk_{i2}) = \left(g^{1/\beta_i}, g_1^{\omega_i}\right)$ as his/her public key.

(6) **Tag-Generation:** On input the private key sk_i, the public key pk_i, the shared data m and the index j. Then u_i generates the tag as follows.

• u_i computes the tag $\sigma = sk_{i1} \cdot (H_2(j) \cdot v^m)^{sk_{i2}}$ for the shared data m.

• u_i computes $P_i = (P_{i1}, P_{i2}) = \left(g_2^{\beta_i} \cdot pk_{i1}, g^{\beta_i}\right)$. Then u_i submits the tag σ_j and P_i to the CSP.

• The validity and correctness of the tag are verified by the equation as follows.

$$\begin{aligned} e\left(\sigma_j, pk_{i1}\right) &= \left(sk_{i1} \cdot \left(H_2(j) \cdot v^m\right)^{sk_{i2}}, pk_{i1}\right) \\ &= e\left(H_1 (ID_i)^{\partial \cdot \omega_i \cdot \beta_i} \cdot \left(H_2(j) \cdot v^m\right)^{\beta_i}, g^{1/\beta_i}\right) \\ &= e\left(H_1 (ID_i), pk_{i2}\right) \cdot e\left(H_2(j) \cdot v^m, g\right). \end{aligned}$$

(7) **Challenge-Generation:** Any group user could send request to the TPA for auditing, ADs shares (ask, apk) with u_i. On input the public keys PK of GUs, the identity information ID of GUs and the index $j_{\max}$. The detailed steps to generate the challenge are described as follows.

i. u_i sends the authorization to the TPA.

• u_i asks the TPA for its identity value ID_T. The TPA encrypts the identity value ID_T with apk to generate $(ID_T)_{apk}$ and sends it to u_i.

• u_i receives the information $(ID_T)_{apk}$ from the TPA, decrypts it with ask to obtain the identity ID_T, generates the authorization $K = (ID_T, t_m)$ and sends K to the TPA, where t_m is the time stamp.

• The TPA checks the correctness of the authorization K. If it is valid, the TPA accepts the authorization of u_i and generates the challenge.

ii. The TPA generates the challenge, and sends it to the CSP.

• The TPA encrypts its identity ID_T with the public key cpk of the CSP to generate $(ID_T)_{cpk}$.

• The TPA randomly chooses the index of shared data $J \subseteq \{1, \ldots, j_{\max}\}$, $\theta \in Z_p$ and $\eta_j \in Z_p$.

• The TPA generates the challenge $\Psi = \left(\overline{ID}, \overline{PK}, \overline{T}\right)$, where $\overline{ID} = \left\{H_1 (ID_i)^{\theta} \,|i = 1, \ldots S\right\}$, $\overline{PK} = \left\{pk_{i1}^{\theta} \,|i = 1, \ldots S\right\}$, $\overline{T} = \left\{\left(j, \eta_j\right) | j \in J\right\}$.

• The TPA sends $R = \left((ID_T)_{cpk}, K, \Psi\right)$ to the CSP.

(8) **Proof-Generation:** On input R, the CSP verifies the authorization, and generates the proof for the challenge. The detailed steps are described as follows.

i. The CSP checks the validity of the authorization.

• The CSP decrypts $(ID_T)_{cpk}$ with its private key csk to obtain the identity value ID_T.

• The CSP decrypts K with ask to obtain the identity value ID'_T.

• If $ID_T = ID'_T$ and the time stamp t_m is valid, the CSP generates the proof. Otherwise the CSP stops interaction.

ii. The CSP generates the proof for the challenge Ψ and sends it to the TPA as follows.

• The CSP chooses $\overline{M} = \{m_j \,|j \in J\}$ and $\Lambda = \{\sigma_j \,|j \in J\}$.

• The CSP computes $\overline{\sigma} = \prod_{j\in J}\left(\frac{e(\sigma_j, pk_{i1}^{\theta})^{\eta_j}}{e(H_1(ID_i)^{\theta}, pk_{i2})^{\eta_j}}\right)$ and $\phi = \sum_{j\in J}\eta_j \cdot m_j$.

• The CSP sends the proof $\Omega = (\overline{\sigma}, \phi)$ to the TPA.

(9) **Proof-Verification:** The TPA checks the correctness of the proof as follows.

• The TPA verifies the following equation. The TPA outputs 1, if

$$\overline{\sigma} = e\left(\prod_{j\in J} H_2(j)^{\eta_j} \cdot v^{\phi}, g^{\theta}\right).$$

Otherwise, the TPA outputs 0.

• The validity and correctness of the proof are verified by the equation as follows.

$$\begin{aligned}\overline{\sigma} &= \prod_{j\in J}\left(\frac{e\left(\sigma_j, pk_{i1}^{\theta}\right)^{\eta_j}}{e\left(H_1\left(ID_i\right)^{\theta}, pk_{i2}\right)^{\eta_j}}\right)\\ &= \prod_{j\in J}\left(\frac{e\left(H_1\left(ID_i\right), pk_{i2}\right)\cdot e\left(H_2(j)\cdot v^{m_j}, g\right)}{e\left(H_1\left(ID_i\right), pk_{i2}\right)}\right)^{\theta\cdot\eta_j}\\ &= e\left(\prod_{j\in J} H_2(j)^{\eta_j}\cdot v^{\phi}, g^{\theta}\right).\end{aligned}$$

B. SUPPORT USER IDENTITY TRACING

For any illegal operation of the malicious user on shared data, at least d group managers could cooperate to trace the malicious user's real identity, which avoids the single-manager from abusing permissions.

• The group managers use the Lagrange interpolation polynomial $F_l(x) = \prod_{0\le r\le d, r\ne l}\frac{x-r}{l-r}$ to build polynomial $y(x) = \sum_{l=1}^{d} f(l)\cdot F_l(x) = \sum_{l=1}^{d} k(l)\cdot F_l(x)$ and compute $k = y(0) = \sum_{l=1}^{d} k(l)\cdot F_l(0)$.

• The group managers calculate $pk_{i1} = P_{i1}/(P_{i2})^k$ to reveal the real identity of the malicious user.

C. SUPPORT USER REVOCATION

When users revoke from the group, their public keys and private keys must be immediately declared invalid, and their tags must be removed. KGC generates a new authorization key pair (ask', apk') to ADs. ADs share (ask', apk') with the existing group users. The user revocation involves three entities. They are the revoked user u_i, the trusted user u_j and the CSP.

• The CSP randomly chooses $\rho \in Z_P$ and sends it to u_j.

• u_j calculates $W_1 = d_{j1}^{d_{j2}}$, $W_2 = \rho\cdot\beta_j$, and sends W_1 and W_2 to u_i.

• u_i computes $R_1 = \left(\frac{W_1}{d_{i1}^{d_{i2}}}\right)^{\beta_i}$, $R_2 = \frac{W_2}{\beta_i}$, and sends R_1 and R_2 to the CSP.

• The CSP computes $R_3 = \frac{R_2}{\rho} = \frac{W_2}{\beta_i\cdot\rho} = \frac{\beta_j}{\beta_i}$.

• After checking all shared data m_{i^*} and tags σ_{i^*} $(1 \le i^* \le j_{\max})$ signed by u_i, the CSP transforms m_{i^*} and σ_{i^*}.

$$\begin{aligned}\sigma'_{i^*} &= (R_1\cdot\sigma_{i^*})^{R_3}\\ &= \left(\left(\frac{W_1}{d_{i1}^{d_{i2}}}\right)^{\beta_i}\cdot d_{i1}^{\beta_i d_{i2}}\cdot\left(H_2\left(i^*\right)\cdot v^{m_{i^*}}\right)^{\beta_i}\right)^{\beta_j/\beta_i}\\ &= \left(H_1\left(ID_j\right)^{\partial}\right)^{\beta_j\cdot d_{j2}}\cdot\left(H_2\left(i^*\right)\cdot v^{m_{i^*}}\right)^{\beta_j}.\end{aligned}$$

σ'_{i^*} is the tag of u_j for shared data m_{i^*}.

• u_j updates P_i.

IV. SECURITY ANALYSIS

The content of this section proves that our proposed scheme satisfies unforgeability and anonymity.

Theorem 1: If the double-BDH hypothesis holds, the proposed scheme satisfies the tag unforgeability under random oracle model.

Proof: It is proved that if the adversary $\mathcal{A}_|$ can win the game I with the probability ε_1 that can't be ignored, after experiencing the most q_{H1} times H_1-Hash-Query, q_{H2} times H_2-Hash-Query, q_{K1} times Partial-Key-Query, q_{K2} times Secret-Value-Query, q_{K3} times Public-Key-Query, q_{Kr} times Public-Key-Replacement-Query and q_σ times Tag-Query, then the challenger $\mathcal{C}$ could find the solution of the double-BDH problem with the probability $\varepsilon'_1 \ge \varepsilon_1/((q_{k1}+q_\sigma)\cdot 2e)$ that can't be ignored. Given an example (PG, g, g^a, g^b, g^c) of the double-BDH problem, the goal of $e(g,g)^{\frac{ac}{b}}$ is computation.

Setup: $\mathcal{C}$ generates the system master key ∂ and the public parameter $param$, sets up $v = g^{b\cdot r_0}$. Then $\mathcal{C}$ sends the public parameter to $\mathcal{A}_|$.

H_1-Hash-Query: $\mathcal{C}$ maintains the H_1 hash list $L_{H_1} = \{(ID, Q, h_1, pk_{ID2}, \Gamma)\}$. $\mathcal{A}_|$ asks $\mathcal{C}$ for any identity ID^* of H_1 hash query. $\mathcal{C}$ looks up the identity ID^* in L_{H_1}. If it does not exist, $\mathcal{C}$ randomly selects $h_1^* \in Z_p$ and tosses coins to select $\Gamma \in \{0,1\}$. The probability of $\Gamma = 0$ is γ, the probability of $\Gamma = 1$ is $1-\gamma$.

• If $\Gamma = 1$, $\mathcal{C}$ sets $Q^* = g^{h_1^*}$, $pk_{ID^*2} = g^{1/h_1^*}$, adds them to L_{H_1}.

• If $\Gamma = 0$, $\mathcal{C}$ sets $Q^* = \left(g^b\right)^{h_1^*}$, $pk_{ID^*2} = (g^a)^{1/h_1^*}$, and adds them to L_{H_1}.

$\mathcal{C}$ finds Q^* in L_{H_1} and sends it to $\mathcal{A}_|$.

H_2-Hash-Query: $\mathcal{C}$ maintains the H_2 hash list $L_{H_2} = \{(j, X, h_2)\}$. $\mathcal{A}_|$ asks $\mathcal{C}$ for the index j^* of H_2 hash query. $\mathcal{C}$ looks up the index j^* in L_{H_2}. If it does not exist, $\mathcal{C}$ randomly selects $h_2^* \in Z_p$, sets $X^* = (g^a)^{h_2^*}$, adds them to L_{H_2}, sends X^* to $\mathcal{A}_|$.

Partial-Key-Query: $\mathcal{C}$ maintains the partial key list $L_p = \{(ID, D_{ID}, \beta_{ID}, pk_{ID1})\}$. $\mathcal{A}_|$ inquires of $\mathcal{C}$ about identity ID^*. $\mathcal{C}$ looks up ID^* and D_{ID^*} in L_p. If ID^* does not exist, $\mathcal{C}$ makes a query on H_1. If D_{ID^*} does not exist, $\mathcal{C}$ does these steps as follows.

• If $\Gamma = 1$, $\mathcal{C}$ stops interaction.

• If $\Gamma = 0$, $\mathcal{C}$ sets $(d_{ID^*1}, d_{ID^*2}) = \left((g^a)^{h_1^*}, \omega_{ID^*}\right)$, where $\omega_{ID^*} \in Z_p$, and adds D_{ID^*} to L_p.

$\mathcal{C}$ finds D_{ID^*} in L_p and sends it as the partial key of ID^* to $\mathcal{A}_{\mid}$.

Secret-Value-Query: $\mathcal{A}_{\mid}$ inquires of $\mathcal{C}$ about the secret value of identity ID^*. $\mathcal{C}$ looks up ID^* and β_{ID^*} in L_p. If ID^* does not exist, $\mathcal{C}$ performs a partial key query. If β_{ID^*} does not exist, $\mathcal{C}$ chooses $\beta_{ID^*} \in Z_p$, sets $pk_{ID^*1} = \left(g^b\right)^{1/\beta_{ID^*}}$, and adds them to L_p. $\mathcal{C}$ finds β_{ID^*} in L_p and sends it as the secret value of ID^* to $\mathcal{A}_{\mid}$. The partial key and the secret value constitute the private key.

Public-Key-Query: $\mathcal{A}_{\mid}$ inquires of $\mathcal{C}$ about the public key of identity ID^*. $\mathcal{C}$ looks up ID^*, pk_{ID^*1} and pk_{ID^*2} in L_{H_1} and L_p. If they do not exist, $\mathcal{C}$ makes a partial key query and a secret value query, finds the public key of identity ID^* in L_p and sends it to $\mathcal{A}_{\mid}$.

Public-Key-Replacement-Query: $\mathcal{C}$ submits the tuple $\left(ID^*, pk'_{ID^*1}, pk'_{ID^*2}\right)$ to $\mathcal{A}_{\mid}$. $\mathcal{C}$ looks up two tuples that contain ID^* in L_p and L_{H_1}. If they exist, $\mathcal{C}$ updates the public key of ID^* to $\left(pk'_{ID^*1}, pk'_{ID^*2}\right)$. Otherwise, $\mathcal{C}$ adds $\left(ID^*, pk'_{ID^*1}, pk'_{ID^*2}\right)$ in L_p and L_{H_1}.

Tag-Query: $\mathcal{A}_{\mid}$ submits (ID^*, j^*, m^*) to $\mathcal{C}$.

- If $\Gamma = 1$, $\mathcal{C}$ stops interaction.
- If $\Gamma = 0$, $\mathcal{C}$ looks up $H_2(j^*)$, partial key, secret value in L_{H_2} and L_p to sign m^*. $\mathcal{C}$ sends the tag to $\mathcal{A}_{\mid}$.

Forge: $\mathcal{A}_{\mid}$ outputs the tag σ' of the data m' on ID' with the public key pk'_{ID}.

Challenge: If $\mathcal{A}_{\mid}$ wins the game I, $\mathcal{C}$ obtains the equation $e\left(\sigma', pk'_{ID1}\right) = e\left(H_1\left(ID'\right), pk'_{ID2}\right) \cdot e\left(H_2\left(j'\right) \cdot v^{m'}, g\right)$.

$\mathcal{C}$ looks up $\left(ID', Q', h'_1, \Gamma'\right)$.

- If $\Gamma^* = 0$, $\mathcal{C}$ stops interaction.
- If $\Gamma^* = 1$, $\mathcal{C}$ sets $H_1\left(ID'\right) = \left(g^b\right)^{h'_1}$, $\overline{ID'} = \left(g^c\right)^{h'_1}$, $pk_{ID'1} = \left(g^b\right)^{1/\beta_{ID'}}$, $\overline{PK'} = \left(g^c\right)^{1/\beta_{ID'}}$ and chooses $\eta_{j'} \in Z_P$. Then $\mathcal{C}$ sends $\Psi = \left(\overline{ID'}, \overline{PK'}, \left(j', \eta_{j'}\right)\right)$ to $\mathcal{A}_{\mid}$. $\mathcal{A}_{\mid}$ sets $\overline{ID'} = H_1\left(ID'\right)^{\theta}$, $\overline{PK'} = \left(pk_{ID'1}\right)^{\theta}$ and $\theta = c/b$.

Response: $\mathcal{A}_{\mid}$ outputs the response $\Omega = \left(\overline{\sigma'}, \phi\right)$. If it satisfies $\overline{\sigma'} = e\left(H_2\left(j'\right)^{\eta_{j'}} \cdot v^{\phi}, g^{\theta}\right)$, the response is valid. $\mathcal{C}$ solves the difficult problem as follows.

$$\left(\frac{\overline{\sigma'}}{e\left(g,g\right)^{c\cdot\phi\cdot r_0}}\right)^{1/\left(h'_2\cdot\eta_{j'}\right)} = \left(\frac{e\left(H_2\left(j'\right)^{\eta_{j'}}\cdot v^{\phi}, g^{\theta}\right)}{e\left(g,g\right)^{c\cdot\phi\cdot r_0}}\right)^{1/\left(h'_2\cdot\eta_{j'}\right)} = \left(\frac{e\left(\left(g^a\right)^{h'_2\cdot\eta_{j'}}, g^{c/b}\right)\cdot e\left(g^{b\cdot\phi\cdot r_0}, g^{c/b}\right)}{e\left(g,g\right)^{c\cdot\phi\cdot r_0}}\right)^{1/\left(h'_2\cdot\eta_{j'}\right)} = e\left(g,g\right)^{\frac{ac}{b}}.$$

In the game I, the possibility of challenger $\mathcal{C}$ and adversary $\mathcal{A}_{\mid}$ stopping interaction only exists in Partial-Key-Query and Tag-Query, thus the probability that $\mathcal{C}$ outputs $e\left(g,g\right)^{\frac{ac}{b}}$ is $\varepsilon'_1 \geq \varepsilon_1 \cdot \gamma \cdot (1-\gamma)^{q_{k1}+q_\sigma} \geq \varepsilon_1 / \left((q_{k1}+q_\sigma)\cdot 2e\right)$.

Theorem 2: If the v-BDH hypothesis holds, the proposed scheme satisfies the tag unforgeability under random oracle model.

Proof: It is proved that if the adversary $\mathcal{A}_{||}$ can win the game II with the probability ε_2 that can't be ignored, after experiencing the most q_{H1} times H_1-Hash-Query, q_{H2} times H_2-Hash-Query, q_{K1} times Secret-Value-Query, q_{K2} times Public-Key-Query and q_σ times Tag-Query, then the challenger $\mathcal{C}$ could find the solution of the BDH problem with the probability $\varepsilon'_2 \geq \varepsilon_2 / \left((q_{k1}+q_\sigma)\cdot 2e\right)$ that can't be ignored. Given an example $\left(G_1, g, g^a, g^b, g^{ac}\right)$ of the BDH problem, the goal of $e\left(g,g\right)^{bc}$ is computation.

Setup: $\mathcal{C}$ generates the system master key ∂ and the public parameter *param*, sets up $v = g^{a\cdot r_0}$. Then $\mathcal{C}$ sends the public parameter to $\mathcal{A}_{||}$.

H_1-Hash-Query: $\mathcal{C}$ maintains the H_1 hash list $L_{H_1} = \{(ID_i, Q, h_1, pk_{i2}, \Gamma)\}$. $\mathcal{A}_{||}$ asks $\mathcal{C}$ for any identity ID^* of H_1 hash query. $\mathcal{C}$ looks up the identity ID^* in L_{H_1}. If it does not exist, $\mathcal{C}$ randomly selects $h_1^* \in Z_p$ and tosses coins to select $\Gamma \in \{0,1\}$. The probability of $\Gamma = 0$ is γ, the probability of $\Gamma = 1$ is $1-\gamma$.

- If $\Gamma = 1$, $\mathcal{C}$ sets $Q^* = g^{h_1^*}$, $pk_{ID^*2} = g^{1/h_1^*}$, and adds them to L_{H_1}.
- If $\Gamma = 0$, $\mathcal{C}$ sets $Q^* = \left(g^a\right)^{h_1^*}$ $pk_{i2^*} = g^{1/h_1^*}$, and adds them to L_{H_1}.

$\mathcal{C}$ finds Q^* in L_{H_1} and sends it to $\mathcal{A}_{||}$.

H_2-Hash-Query: $\mathcal{C}$ maintains the H_2 hash list $L_{H_2} = \{(j, X, h_2)\}$. $\mathcal{A}_{||}$ asks $\mathcal{C}$ for the index j^* of H_2 hash query. $\mathcal{C}$ looks up the index j^* in L_{H_2}. If it does not exist, $\mathcal{C}$ randomly selects $h_2^* \in Z_p$, sets $X^* = \left(g^b\right)^{h_2^*}$, adds them to L_{H_2}, sends X^* to $\mathcal{A}_{||}$.

Secret-Value-Query: $\mathcal{C}$ maintains the secret value list $L_s = \{(ID, \beta_{ID}, pk_{ID1}, \Gamma)\}$. Because $\mathcal{A}_{||}$ has the system master key, the partial key is no longer queried. $\mathcal{A}_{||}$ inquires of $\mathcal{C}$ about the secret value of identity ID^*. $\mathcal{C}$ looks up ID^* and β_{ID^*} in L_s. If ID^* does not exist, $\mathcal{C}$ performs a query on H_1. If β_{ID^*} does not exist, $\mathcal{C}$ chooses $\beta_{ID^*} \in Z_p$.

- If $\Gamma = 1$, $\mathcal{C}$ sets $pk_{ID^*1} = \left(g^a\right)^{1/\beta_{ID^*}}$ and adds it to L_s, then stops interaction.
- If $\Gamma = 0$, $\mathcal{C}$ sets $pk_{ID^*1} = g^{1/\beta_{ID^*}}$ and adds it to L_s.

$\mathcal{C}$ finds β_{ID^*} and sends it as the secret value of ID^* to $\mathcal{A}_{||}$.

Public-Key-Query: $\mathcal{A}_{||}$ inquires of $\mathcal{C}$ about the public key of identity ID^*. $\mathcal{C}$ looks up ID^*, pk_{ID^*1} and pk_{ID^*2} in L_{H_1} and L_s. If they do not exist, $\mathcal{C}$ makes a secret value query. $\mathcal{C}$ finds the public key of identity ID^* in L_s and sends it to $\mathcal{A}_{\mid}$.

Tag-Query: $\mathcal{A}_{||}$ submits (ID^*, j^*, m^*) to $\mathcal{C}$.

- If $\Gamma = 1$, $\mathcal{C}$ stops interaction.
- If $\Gamma = 0$, $\mathcal{C}$ looks up $H_2(j^*)$, secret value in L_{H_2} and L_s to sign m^*. $\mathcal{C}$ sends the tag to $\mathcal{A}_{||}$.

Forge: $\mathcal{A}_{||}$ outputs the tag σ' of the data m' on ID'.

Challenge: If $\mathcal{A}_{||}$ wins the game II, $\mathcal{C}$ obtains the equation $e\left(\sigma', pk'_{ID1}\right) = e\left(H_1\left(ID'\right), pk'_{ID2}\right) \cdot e\left(H_2\left(j'\right) \cdot v^{m'}, g\right)$.

$\mathcal{C}$ looks up $\left(ID', Q', h'_1, T'\right)$.

• If $\Gamma^* = 0$, $\mathcal{C}$ stops interaction.

• If $\Gamma^* = 1$, $\mathcal{C}$ sets $H_1\left(ID'\right) = \left(g^a\right)^{h'_1}$, $\overline{ID'} = \left(g^{ac}\right)^{h'_1}$, $pk_{ID'1} = \left(g^a\right)^{1/\beta_{ID'}}$, $\overline{PK'} = \left(g^{ac}\right)^{1/\beta_{ID'}}$, and chooses $\eta_{j'} \in Z_P$. Then $\mathcal{C}$ sends $\Psi = \left(\overline{ID'}, \overline{PK'}, \left(j', \eta_{j'}\right)\right)$ to $\mathcal{A}_{||}$. $\mathcal{A}_{||}$ sets $\overline{PK'} = \left(pk_{ID'1}\right)^{\theta}$, $\overline{ID'} = H_1\left(ID'\right)^{\theta}$ and $\theta = c$.

Response: $\mathcal{A}_{||}$ outputs the response $\Omega = \left(\overline{\sigma'}, \phi\right)$. If it satisfies $\overline{\sigma'} = e\left(H_2\left(j'\right)^{\eta_{j'}} \cdot v^{\phi}, g^{\theta}\right)$, the response is valid. $\mathcal{C}$ solves the difficult problem as follows.

$$\begin{aligned}
&\left(\frac{\overline{\sigma'}}{e\left(g,g\right)^{a\cdot c\cdot\phi\cdot r_0}}\right)^{1/\left(h'_2\cdot\eta_{j'}\right)} \\
&= \left(\frac{e\left(H_2\left(j'\right)^{\eta_{j'}}\cdot v^{\phi}, g^{\theta}\right)}{e\left(g,g\right)^{a\cdot c\cdot\phi\cdot r_0}}\right)^{1/\left(h'_2\cdot\eta_{j'}\right)} \\
&= \left(\frac{e\left(\left(g^b\right)^{h'_2\cdot\eta_{j'}}, g^c\right)\cdot e\left(g^{a\cdot\phi\cdot r_0}, g^c\right)}{e\left(g,g\right)^{a\cdot c\cdot\phi\cdot r_0}}\right)^{1/\left(h'_2\cdot\eta_{j'}\right)} \\
&= e\left(g,g\right)^{bc}.
\end{aligned}$$

In the game II, the possibility of challenger $\mathcal{C}$ and adversary $\mathcal{A}_{||}$ stopping interaction only exists in Secret-Key-Query and Tag-Query, so the probability that $\mathcal{C}$ outputs $e\left(g,g\right)^{bc}$ is $\varepsilon'_2 \geq \varepsilon_2 \cdot \gamma \cdot (1-\gamma)^{q_{k1}+q_\sigma} \geq \varepsilon_2 / \left((q_{k1}+q_\sigma)\cdot 2e\right)$.

Theorem 3: The possibility of that any adversary $\mathcal{A}_{|||}$ obtains the user's identity in the process of auditing is 0.

Anonymity guarantees that two different users generate different tags, but the calculated responses are same. Therefore, after receiving the response, $\mathcal{A}_{|||}$ could not distinguish which user uploads the data.

Given the equation $(m, \Psi) = \left(m, j, \eta_j, \overline{ID}, \overline{PK}\right)$, where $\overline{ID} = \left\{H_1\left(ID_i\right)^{\theta} | i = 1, \ldots S\right\}$ and $\overline{PK} = \left\{pk_{i1}^{\theta} | i = 1, \ldots S\right\}$. Challenger $\mathcal{C}$ randomly selects two users ID_i and $ID_{i'}$, then computes their tags σ_i and $\sigma_{i'}$, proofs Ω_i and $\Omega_{i'}$ for m as follows.

$$\phi = \eta_j \cdot m_j, \quad \overline{\sigma} = \frac{e\left(\sigma_j, pk_{i1}^{\theta}\right)^{\eta_j}}{e\left(H_1\left(ID_i\right)^{\theta}, pk_{i2}\right)^{\eta_j}}.$$

$$\phi' = \eta_j \cdot m_j, \quad \overline{\sigma'} = \frac{e\left(\sigma_j, pk_{i1'}^{\theta}\right)^{\eta_j}}{e\left(H_1\left(ID_{i'}\right)^{\theta}, pk_{i2'}\right)^{\eta_j}}.$$

$$\begin{aligned}
\overline{\sigma} &= \frac{e\left(\sigma_j, pk_{i1}^{\theta}\right)^{\eta_j}}{e\left(H_1\left(ID_i\right)^{\theta}, pk_{i2}\right)^{\eta_j}} \\
&= \left(\frac{e\left(H_1\left(ID_i\right), pk_{i2}\right)\cdot e\left(H_2\left(j\right)\cdot v^{m_j}, g\right)}{e\left(H_1\left(ID_i\right), pk_{i2}\right)}\right)^{\theta\cdot\eta_j} \\
&= e\left(H_2\left(j\right)^{\eta_j}\cdot v^{\phi}, g^{\theta}\right).
\end{aligned}$$

$$\begin{aligned}
\overline{\sigma'} &= \frac{e\left(\sigma_j, pk_{i1'}^{\theta}\right)^{\eta_j}}{e\left(H_1\left(ID_{i'}\right)^{\theta}, pk_{i2'}\right)^{\eta_j}} \\
&= \left(\frac{e\left(H_1\left(ID_{i'}\right), pk_{i2'}\right)\cdot e\left(H_2\left(j\right)\cdot v^{m_j}, g\right)}{e\left(H_1\left(ID_{i'}\right), pk_{i2'}\right)}\right)^{\theta\cdot\eta_j} \\
&= e\left(H_2\left(j\right)^{\eta_j}\cdot v^{\phi}, g^{\theta}\right).
\end{aligned}$$

From the above formulas, we can get $\phi = \phi'$ and $\overline{\sigma} = \overline{\sigma'}$.

Therefore, the adversary $\mathcal{A}_{|||}$ cannot obtain any information related to the identity of the user. Our scheme satisfies the anonymity.

V. EFFICIENCY ANALYSIS

In this section, we measure the performance in terms of functionality, communication cost and computation cost from our scheme.

A. FUNCTIONALITY COMPARISON

Table 1 lists the functions of our scheme compared with the scheme [30] and the scheme [36] for shared data in the cloud. The scheme [30] is a privacy-preserving cloud data integrity verification scheme with multiple users based on certificateless. The scheme [36] is a certificateless integrity verification scheme of the shared data on cloud storage. As Table 1 shows, the related schemes do not have the important properties of authorized auditing, traceability and non-frameability (ADs can guarantee the fairness of identity tracing). Furthermore, our scheme also can support user revocation from group and user privacy-preserving. Hence, our scheme has a more extensive application.

B. PERFORMANCE ANALYSIS

The performance analysis is mainly measured from communication cost and computational cost. For the notations used in our scheme, let T_{exp} represents the cost of one exponential operation, T_{mul} represents the cost of one multiplication operation, T_{pair} represents the cost of one pairing operation, n represents the total number of the users, z represents the number of data blocks for the shared file, d represents the number of the group user subsets used for challenge, and j represents the number of data blocks used for challenge in the process of data auditing. We do not consider the cost of the general hash function, pseudo-random number generation, because their costs are negligible. The analysis results are as follows.

1) COMMUNICATION COST

The communication cost of our scheme mainly involves two aspects: challenge and proof. In the data auditing phase, the TPA submits the challenge $R = \left((ID_T)_{cpk}, K, \Psi\right)$ to the CSP, its size is $(2j+2)\,|q| + 2n\,|G_1|$. The CSP sends the proof $\Omega = (\overline{\sigma}, \phi)$ to the TPA, its size is $2\,|q|$. The total communication cost of our scheme is $(2j+4)\,|q| + 2n\,|G_1|$. To support authorized auditing, our scheme adds $(ID_T)_{cpk}$ and K in the challenge, which brings an additional cost of $2\,|q|$ compared with scheme [30]. Compared with total communication cost, the additional overhead of $2\,|q|$ is low and acceptable. The specific comparison is shown in Table 2.

TABLE 1. Comparison of function.

Main Function	Authorized process	User Revocation	Identity Privacy	Traceability	Non-frameability
Scheme [30]	No	No	Yes	No	No
Scheme [36]	No	Yes	No	No	No
Ours	Yes	Yes	Yes	Yes	Yes

TABLE 2. Comparison of communication cost.

Scheme	Challenge	Proof
Scheme [30]	$2j\|q\|+2n\|G_1\|$	$2\|q\|$
Scheme [36]	$3\|q\|$	$(d+3)\|q\|+d\|G_1\|$
Ours	$(2j+2)\|q\|+2n\|G_1\|$	$2\|q\|$

TABLE 3. Comparison of computation cost.

Scheme	Tag Phase	Data Auditing Phase
Scheme [30]	$4T_{pair}+3T_{exp}+3T_{mul}$	$4T_{pair}+(2n+3j+3)T_{exp}+3jT_{mul}$
Scheme [36]	$3zT_{pair}+3zT_{exp}+zT_{mul}$	$(d+2)T_{pair}+(2j+d)T_{exp}+2(j+d)T_{mul}$
Ours	$3T_{pair}+5T_{exp}+4T_{mul}$	$3T_{pair}+(2n+3j+1)T_{exp}+(3j-1)T_{mul}$

TABLE 4. Time of main phases of the scheme.

Phase	Parameters and Key Generation	Tag Generation	Tag Verification	Challenge and Proof Generation	Proof Verification
Scheme [30] (s)	0.058734	0.018616	0.046471	0.049412	0.034735
Ours (s)	0.042986	0.012843	0.038389	0.048558	0.016594

2) COMPUTATION COST

We could find that the computation cost of our scheme is more efficient compared with the scheme [30] and the scheme [36] in Table 3. Because T_{pair} is the highest cost operation, our scheme has the fewest T_{pair}. Although in the **Tag-Generation** phase, our scheme adds P_i as the partial tag to satisfy traceability, which increases the cost $(2T_{exp}+T_{mul})$ compared with scheme [30], the total computation cost is still the lowest. In contrast, the scheme [36] divides the same shared data into z blocks to sign, which makes it have more T_{pair}. In the **Proof-Verification** phase, our scheme has the shorter length of the proof than scheme [30], which reduces the computation cost.

C. EXPERIMENTAL RESULTS

In the experiment, we use Pairing Based Cryptography (PBC) to simulate the operations of this scheme. All tests are applied to a Windows system with an Intel Core i7 CPU processor running at 3.60 GHz and 8GB RAM. Every result is an average of 10 tests. The first experiment is to calculate the cost of generating tags in our scheme. We set 50 group users, and the amount of shared data ranges from 1000 to 10000. As Figure 2 shows, the cost of generating tags has a linear relationship with the number of shared data. However, generating tags for 1000 shared data only spends about 193 seconds, which could be accepted by users. Moreover, a shared

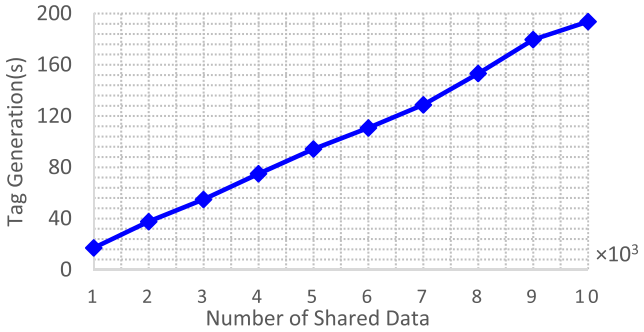

FIGURE 2. Computation cost of tag generation.

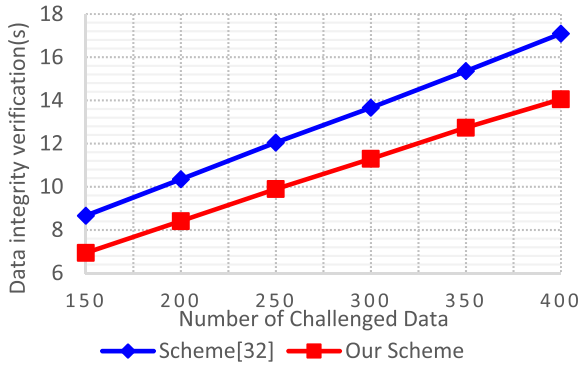

FIGURE 3. Time of data auditing phase.

data file usually executes one **Tag-Generation** algorithm, which has little impact on the data auditing.

Then we compare the main phases cost of our scheme and the scheme [30]. As Table 4 shows, our scheme is more efficient than the scheme [30]. The specific result is shown in Figure 2.

We compare the time spent in the data auditing phase of the scheme in this paper with the scheme [30]. The specific result is shown in Figure 3. We could discover that the time used for data auditing increases linearly to the number of challenged data files. The scheme [30] takes about 17s to finish one data auditing of 400 data blocks. However, our scheme only spends 14s. Furthermore, the auditing time of the scheme [30] has a faster growth rate compared with the scheme in this paper. As the number of challenged data increases, the gap between our scheme and the scheme [30] could become bigger. Therefore, in the process of data auditing, our scheme has less computational overhead than the scheme [30].

VI. CONCLUSIONS

In this paper, we propose a privacy-preserving cloud auditing scheme for multiple users with authorization and traceability. This scheme not only has the advantage of certificateless signature but also meets the security requirements for cloud auditing of shared data. During the process of data auditing, the TPA cannot get the identity of the group user who uploaded the data. Meanwhile, multiple managers in the proposed scheme could reveal the identity of the malicious user cooperatively, which guarantees the fairness of tracing. Besides, we introduce an identity authorization process between the TPA and the CSP, which protects CSP from malicious harassment. Moreover, our scheme supports the effective revocation of group users, which reduces communication costs. Our scheme is proved efficient by the analysis results.

REFERENCES

[1] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio "Security issues in cloud environments: A survey," *Int. J. Inf. Secur.*, vol. 13, no. 2, pp. 113–170, Apr. 2014.
[2] W. Hsien, C. Yang, and M. Hwang, "A survey of public auditing for secure data storage in cloud computing," *Int. J. Netw. Secur.*, vol. 18, no. 1, pp. 133–142, 2016.
[3] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1167–1179, Jun. 2015.
[4] M. S. Hwang and C. C. Lee, "Research issues and challenges for multiple digital signatures," *Int. J. Netw. Secur.*, vol. 1, no. 1, pp. 1–7, Jul. 2005.
[5] M. S. Hwang, C. C. Lee, and Y. L. Tang, "Two simple batch verifying multiple digital signatures," in *Proc. ICICS*, X'ian, China, 2001, pp. 233–237.
[6] B. Wang, H. Li, and M. Li, "Privacy-preserving public auditing for shared cloud data supporting group dynamics," in *Proc. ICC*, Budapest, Hungary, Jun. 2013, pp. 1946–1950.
[7] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. CCS*, Alexandria, VA, USA, 2007, pp. 598–609.
[8] H. Shacham and B. Waters, "Compact proofs of retrievability," *J. Cryptol.*, vol. 26, no. 3, pp. 442–483, Jul. 2013.
[9] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. SecureComm*, 2008, pp. 1–10.
[10] F. Sebé, J. Domingo-Ferrer, A. Martínez-Ballesté, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1034–1038, Aug. 2008.
[11] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 847–859, May 2011.
[12] B. Wang, B. Li, H. Li, and F. Li, "Certificateless public auditing for data integrity in the cloud," in *Proc. CNS*, Washington, DC, USA, Oct. 2013, pp. 136–144.
[13] K. Loheswaran and J. Premalatha, "Renaissance system model improving security and third party auditing in cloud computing," *Wireless Pers. Commun.*, vol. 90, no. 2, pp. 1051–1066, Sep. 2016.
[14] B. S. Rawal, V. Vijayakumar, G. Manogaran, R. Varatharajan, and N. Chilamkurti, "Secure disintegration protocol for privacy preserving cloud storage," *Wireless Pers. Commun.*, vol. 103, no. 2, pp. 1161–1177, Nov. 2018.
[15] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. EUROCRYPT*, Berlin, Germany, 1984, pp. 47–53.
[16] H. Wang, J. Domingo-Ferrer, B. Qin, and Q. Wu, "Identity-based remote data possession checking in public clouds," *IET Inf. Secur.*, vol. 8, no. 2, pp. 114–121, Mar. 2014.
[17] 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.
[18] J. Li, H. Yan, and Y. Zhang, "Identity-based privacy preserving remote data integrity checking for cloud storage," *IEEE Syst. J.*, early access, Mar. 18, 2020, doi: 10.1109/jsyst.2020.2978146.
[19] X. Zhang, J. Zhao, L. Mu, Y. Tang, and C. Xu, "Identity-based proxy-oriented outsourcing with public auditing in cloud-based medical cyber–physical systems," *Pervas. Mobile Comput.*, vol. 56, pp. 18–28, May 2019.
[20] X. Zhang, H. Wang, and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," *Inf. Sci.*, vol. 472, pp. 223–234, Jan. 2019.
[21] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. ASIACRYPT*, Berlin, Germany, 2003, pp. 452–473.
[22] C.-T. Li, C.-Y. Weng, and C.-C. Lee, "A secure RFID tag authentication protocol with privacy preserving in telecare medicine information system," *J. Med. Syst.*, vol. 39, no. 8, pp. 1–8, Aug. 2015.

[23] M.-S. Hwang, T.-H. Sun, and C.-C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service," *J. Circuits, Syst. Comput.*, vol. 26, no. 5, May 2017, Art. no. 1750072.
[24] L. Huang, G. Zhang, and A. Fu, "Privacy-preserving public auditing for non-manager group shared data," *Wireless Pers. Commun.*, vol. 100, no. 4, pp. 1277–1294, Jun. 2018.
[25] B. Wang, B. Li, and H. Li, "Knox: Privacy-preserving auditing for shared data with large groups in the cloud," in *Proc. ACNS*, Berlin, Germany, 2012, pp. 507–525.
[26] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 43–56, Jan. 2014.
[27] C.-C. Lee, M.-S. Hwang, and W.-P. Yang, "A new blind signature based on the discrete logarithm problem for untraceability," *Appl. Math. Comput.*, vol. 164, no. 3, pp. 837–841, May 2005.
[28] K. Gu, L. Y. Wang, N. Wu, and N. D. Liao, "Traceable certificateless ring signature scheme for no full anonymous applications," *Int. J. Netw. Secur.*, vol. 20, no. 4, pp. 442–483, 2018.
[29] L. Zhang, F. Zhang, and W. Wu, "A provably secure ring signature scheme in certificateless cryptography," in *Proc. ProvSec*, Berlin, Germany, 2007, pp. 103–121.
[30] 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, 2019.
[31] B. Waters, "Efficient identity-based encryption without random oracles," in *Proc. EUROCRYPT*, Berlin, Germany, 2005, pp. 114–127.
[32] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute based encryption with privacy protection and accountability for CloudIoT," *IEEE Trans. Cloud Comput.*, early access, Feb. 19, 2020, doi: 10.1109/tcc.2020.2975184.
[33] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1717–1726, Aug. 2015.
[34] 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. NSS*, X'ian, China, 2014, pp. 28–40.
[35] X. Yang, Y. Li, J. Wang, T. Ma, and C. Wang, "Revocable identity-based proxy re-signature scheme in the standard model," *J. Commun.*, vol. 40, no. 5, pp. 153–162, 2019.
[36] J. Li, H. Yan, and Y. Zhang, "Certificateless public integrity checking of group shared data on cloud storage," *IEEE Trans. Services Comput.*, early access, Jan. 8, 2018, doi: 10.1109/TSC.2018.2789893.
[37] C. Liu, J. Chen, L. T. Yang, X. Zhang, C. Yang, R. Ranjan, and K. Rao, "Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2234–2244, Sep. 2014.

XIAODONG YANG (Member, IEEE) received the B.S. degree in mathematics from Northwest Normal University, China, in 2002, the M.S. degree in cryptography from Tongji University, China, in 2005, and the Ph.D. degree in cryptography from Northwest Normal University, in 2010. He is currently a Postdoctoral Fellow with the State Key Laboratory of Cryptology of China and a Professor in information and computer science with Northwest Normal University. His research interests include applied cryptography, network security, and cloud computing security. He is also a member of the Chinese Cryptology and Information Security Association.

MEIDING WANG received the B.S. degree from Northwest Normal University, Lanzhou, China, in 2018, where she is currently pursuing the master's degree in computer science. Her current research interest includes cloud computing security.

TING LI received the B.S. degree from Zhengzhou Normal University, Zhengzhou, China, in 2018. She is currently pursuing the master's degree in computer science with Northwest Normal University. Her current research interests include network security, blockchain technology, and their applications.

RUI LIU received the B.S. degree from Henan Polytechnic University, Jiaozuo, China, in 2017. She is currently pursuing the master's degree in computer science with Northwest Normal University. Her current research interest includes vehicular ad hoc network technology.

CAIFEN WANG received the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2003. She is currently a Professor in computer science with Shenzhen Technology University. Her current research interests include network security, cryptographic protocols, and security engineering. She is also a member of the Chinese Cryptology and Information Security Association.