Traceable Dynamic Public Auditing with Identity Privacy Preserving for Cloud Storage

Yinghui Zhang[1,2*], Tiantian Zhang[1], Rui Guo[1], Shengmin Xu[3], Dong Zheng[1,2]
[1] National Engineering Laboratory for Wireless Security,
Xi'an University of Posts and Telecommunications, Xi'an 710121, P.R. China
[e-mail: yhzhaang@163.edu, ttzhng@163.com, guorui@xupt.edu.cn]
[2] Westone Cryptologic Research Center, Beijing 100070, P.R. China
[e-mail: zhengdong@xupt.edu.cn]
[3] Secure Mobile Centre, School of Information Systems,
Singapore Management University, Singapore 178902
[e-mail: smxu@smu.edu.sg]
*Corresponding author: Yinghui Zhang

Received January 19, 2019; revised May 27, 2019; accepted June 21, 2019; published November 30, 2019

Abstract

In cloud computing era, an increasing number of resource-constrained users outsource their data to cloud servers. Due to the untrustworthiness of cloud servers, it is important to ensure the integrity of outsourced data. However, most of existing solutions still have challenging issues needing to be addressed, such as the identity privacy protection of users, the traceability of users, the supporting of dynamic user operations, and the publicity of auditing. In order to tackle these issues simultaneously, in this paper, we propose a traceable dynamic public auditing scheme with identity privacy preserving for cloud storage. In the proposed scheme, a single user, including a group manager, is unable to know the signer's identity. Furthermore, our scheme realizes traceability based on a secret sharing mechanism and supports dynamic user operations. Based on the security and efficiency analysis, it is shown that our scheme is secure and efficient.

Keywords: Identity privacy preserving, traceability, public auditing, user revocation, cloud storage

1. Introduction

As the amount of user data continues to increase, storing these data will bring a huge burden on users with limited resources. Considering storage overhead, more and more organizations and individuals store their data in the cloud [1]. At present, cloud storage is a commonly used tool in various cloud services. Data storage in the cloud has gradually become an irresistible trend [2]. Recently, the development of cloud computing has promoted some simple storage services [3], such as on-line data backup services of Amazon and cloud based software Google Drive. However, due to hardware and software problems or improper operation of the user, the data stored in the cloud server may be changed. It is necessary to ensure the authenticity of data before analyzing and processing it [4], [5]. In order to ensure that the integrity of the data stored in the cloud server is not threatened, many integrity auditing schemes have been proposed [2], [6]-[12].

In a realistic scenario, users may join the cloud storage system at any time, or may leave the system at any time for some reasons. So it is especially important to support flexible user dynamic operations. But some proposed schemes [12]-[14] do not support user dynamic operations. In addition, in some cases, people are reluctant to disclose their identity information to others. So more and more anonymous applications have appeared, such as electronic auctions, electronic voting, etc. Schemes [15] and [16] can be used as possible solutions to achieve privacy protection. Wang et al. proposed anonymous two-factor authentication in distributed systems [17] and in wireless sensor networks [18]. Zhang et al. proposed two blockchain-based fair payment protocols called BPay [19] and BCPay [20] for outsourcing services in cloud computing. The protocol BPay [19] is compatible with the Bitcoin blockchain based on an iterative all-or-nothing checking-proof protocol and a top-down checking method. However, the performance remains to be improved. At the cost of losing the compatibility with the Bitcoin blockchain, the protocol BCPay [20] realizes robust fair payment based on a one round all-or-nothing checking-proof protocol and hence is very efficient in terms of the computation cost and the number of transactions. These two protocols can be adopted to enable trustworthy keyword search over cloud encrypted data with two-side verifiability [21], secure outsourcing computation, and dynamic data integrity in cloud computing. There is a growing demand for storage services. It not only needs to support user dynamic operations, but also needs to ensure that the user's identity privacy is not disclosed. Some proposed schemes [2], [8], [22] do not support identity privacy protection. Therefore, it makes sense to consider how to design a data integrity auditing scheme that has identity privacy protection and supports user dynamic operations.

Cloud storage is one of the hot spots of cloud computing research in recent years [22]-[24]. In 2007, Athenians et al. [25] first proposed the concept of public auditing, which has been widely accepted by researchers. Some improved data integrity checking schemes have been proposed [6], [12], [26]. Shacham et al. [26] proposed compact proofs of retrievability by making use of publicly verifiable homomorphic authenticators from BLS signature [27]. Yu et al. [6] proposed an identity-based remote data integrity checking scheme with perfect data privacy preserving for cloud storage. However, scheme [6] could neither guarantee identity privacy protection nor support efficient dynamic user operation. Wang et al. [12] proposed a privacy-preserving public auditing scheme for shared data in the cloud. Because scheme [12] uses ring signatures, dynamic user operations are not supported. Huang et al. [14] proposed a

privacy-preserving public auditing scheme for non-manager group shared data, but this scheme did not realize dynamic user operation.

In addition, Hu et al. [28] pointed out that currently there is no scheme to realize the public integrity checking with identity privacy protection of efficient dynamic groups in cloud storage, so they proposed an identity-preserving public integrity checking scheme with dynamic groups for cloud storage. They used BLS short signature [29] to verify the legality of user identity, the identity of the signer in this scheme is normally kept confidential. In order to reveal the identity of the signer in the group when necessary, Hu et al. [28] introduced zero-knowledge proof into the signature algorithm in the proposed scheme, so as to realize the group manager to track the identity of the signer. However, the complex zero-knowledge proof reduces the signature efficiency and is difficult to implement. Recently, cloud computing security and privacy has become more and more important [30]-[31]. Zheng and Li et al. proposed a signature scheme [32]-[33] for threshold tracking based on the idea that secret sharing scheme [34] can reconstruct secret. Huang et al. [14] used the threshold tracking method to achieve the identity tracking of signers. Inspired by scheme [14], we introduced the idea of threshold tracking into the signature algorithm in our scheme in order to track the identity of the signer. Since our scheme uses threshold tracking method, a certain number of users can track the identity of signer, rather than relying only on group manager, which is more fair and equitable. In addition, we use the polynomial function proposed by [28] and [35] to construct the group secret key and efficiently update the group secret key.

In this paper, we propose a traceable dynamic public auditing scheme with identity privacy preserving for cloud storage. Our scheme has many attractive features as below:

(1) It realizes data integrity checking, supports dynamic user operation, and enables group secret key update by constructing polynomial functions.

(2) It realizes full anonymity such that a single user including the group manager is unable to know the identity of the user. The cloud server can verify the legitimacy of the user's identity even if it does not know the identity of the user, thereby hiding the identity of the user.

(3) In our scheme, a (*t*, *n*) secret sharing scheme is used to perform user tracking by reconstructing the public key of the signer. Specifically, any subset can reveal the identity of the signer if and only if it consists of at least *t* users.

The rest of the paper is organized as follows: In section 2, we present some relevant preliminaries. In section 3, we present the system model and relevant definitions. We describe our concrete scheme in section 4. In section 5 and section 6, we give security analysis and performance analysis, respectively. Finally, we give concluding remarks in section 7.

2. Preliminaries

2.1 Bilinear Pairing

Let G_1, G_T be two groups of prime order q, and g_1 be a generator of group G_1. e is a bilinear map $e: G_1 \times G_1 \to G_T$, which satisfies the following properties:

(1) For all P, $Q \in G_1$ and $a \in Z_q^*$, $e(P^a, Q) = e(P, Q^a)$.

(2) There is non-degenerate, $e(g_1, g_1) \neq 1$.

2.2 Computational Diffie-Hellman (CDH) Problem

For $x \in Z_q^*$, given g_1, $g_1^x \in G_1$ and $h \in G_1$ as input, output $h^x \in G_1$.

2.3 Discrete Logarithm Problem

For $x \in Z_q^*$, given g_1, $g_1^x \in G_1$ as input, output x.

2.4 BLS Scheme

The BLS signature scheme is described as follows. Signer randomly selects $x \in Z_q^*$ as the private key and calculates $y = g_1^x \in G_1$ as the public key. Then the signature operation is performed by calculating $h = H(M) \in G_1$ and $\sigma = h^x \in G_1$, where $M \in \{0,1\}^*$, $H: \{0,1\}^* \to G_1$ is a hash function and σ is the generated signature of M. Finally signature verification operation is performed by calculating $h = H(M) \in G_1$ and determining Whether the equation $e(h, y) = e(\sigma, g_1)$ holds or not. If the above equation holds, the verification succeeds, otherwise the verification fails.

2.5 Secret Sharing

In a (t, n) threshold scheme [34] based on Lagrange interpolation formula, there are n participants. A $t-1$ degree polynomial $f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$ is constructed, where $a_i \in Z_q^*$ and $1 < t < n$. If the secret s is $f(0)$, n sub-keys are $f(i)$, $1 \le i \le n$. Any t sub-keys can reconstruct polynomial $f(x)$ to obtain secret s. They use $\{(i, f(i))\}_{1 \le i \le t}$ to construct polynomial based on Lagrange interpolation formula:

$$f(x) = \sum_{i=1}^{t} f(i) \prod_{\substack{j=1 \\ j \ne i}}^{t} \frac{(x-j)}{i-j} (\mathrm{mod}\, q) \tag{1}$$

Therefore, without knowing $f(x)$, participants can get secret $s = f(0)$ as follows:

$$s = \sum_{i=1}^{t} f(i) \prod_{\substack{l=1 \\ l \ne j}}^{t} \frac{j}{j-i} (\mathrm{mod}\, q) \tag{2}$$

3. System Model and Definition

In this section, we present the system model, design goals, definition of algorithm, and security problems that might exist.

3.1 System Model

The system model consists of four entities: a cloud storage server, a third party auditor (TPA), a group manager and users. The system model is illustrated in **Fig. 1**. The group manager is mainly responsible for the generation of the user's initial key and the update of the group secret key, and supports the user's dynamic operation. Users joining the system are considered as legitimate users and can store data on the cloud storage server. A certain number of legitimate users can verify the identity of the signer together. The cloud storage server is mainly

responsible for storing user data. TPA is primarily responsible for verifying the integrity of stored data.

During the data integrity checking process, the user sends an integrity checking request to TPA. After receiving the request, TPA generates a challenge message and sends it to the cloud storage server. After receiving the challenge message, the cloud storage server generates corresponding proof information and sends it to TPA. TPA verifies the proof and returns the verification result to the user.

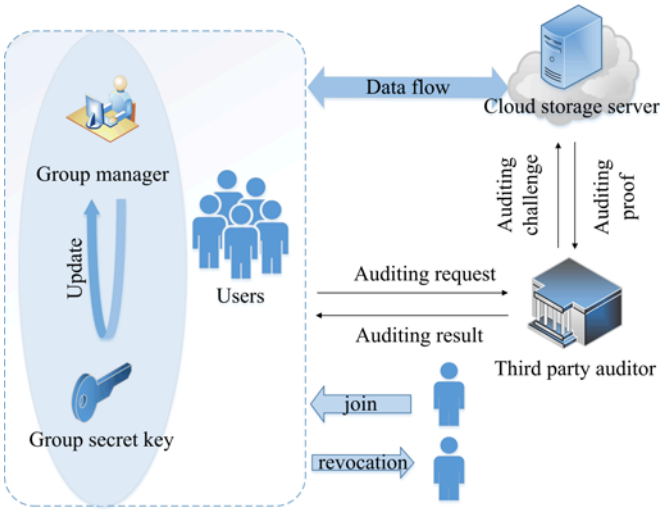

Fig. 1. System model

3.2 Design Goals

(1) Public integrity verification: A public verifier can detect data integrity without retrieving complete storage data.
(2) Full anonymity: Whether in the process of uploading data or in the process of integrity verification, it can ensure that the user's identity privacy is not disclosed.
(3) Traceability: A certain number of legitimate users can jointly track the identity of the signer.
(4) Dynamic user operation: It can efficiently implement user revocation.

3.3 Definition of Algorithm

Our scheme includes eight algorithms: Setup Algorithm, Users Join Algorithm, Group Key Generation Algorithm, Signature Generation Algorithm, Upload Algorithm, Public Integrity Checking Algorithm, Trace Algorithm and Revocation Algorithm.

(1) Setup Algorithm: the setup algorithm is run by group manager. It takes as input security parameter. It outputs public parameter $param$ and issuing key ik of group manager.

(2) Users Join Algorithm: the users join algorithm is run by group manager and user. It takes as input public parameter $param$, issuing key ik of group manager and user's identity ID_k. It outputs the private key sk_k and public key pk_k of user, as well as the user's initial key A_k generated by the group manager.

(3) Group Key Generation Algorithm: the group key generation algorithm is run by group manager. It takes as input public parameter $param$, the user's initial key A_k generated by the group manager. It outputs the group secret key K_g, the group public key ρ, the polynomial

function $f(x)$ and EK.

(4) Signature Generation Algorithm: the signature generation algorithm is run by user. It takes as input public parameter $param$, the user's initial key A_k generated by the group manager, user's private key sk_k,file data M, as well as the public key $\{pk_i\}_{1\le i\le n}$ of the legitimate user in the system. It outputs the metadata information of user data including θ, $\{\sigma_y\}_{1\le y\le m}$, $\{c_y\}_{1\le y\le m}$ and secret sharing $share$.

(5) Upload Algorithm: the upload algorithm is run by user and cloud storage server. It takes as input public parameter $param$, the metadata information of user data including θ, $\{\sigma_y\}_{1\le y\le m}$, $\{c_y\}_{1\le y\le m}$ and secret sharing $share$. Cloud storage server outputs "accept" or "reject".

(6) Public Integrity Checking Algorithm: the public integrity checking algorithm is run by the cloud server and TPA. It takes as input public parameter $param$ and file name $filename$. It outputs "1" or "0" and returns it to the user.

(7) Trace Algorithm: the trace algorithm is run by t valid users. It takes as input public parameter $param$, secret sharing $share$ and the private keys $\{sk_i\}_{1\le i\le t}$ of t valid users. It outputs the public key pk_k of the signer.

(8) Revocation Algorithm: the revocation algorithm is run by group manager. It takes as input public parameter $param$, the polynomial function $f(x)$ and the identity of the revoked user ID_k. It outputs new group secret key K_g', new group public key ρ', new polynomial function $f'(x)$ and EK'.

3.4 Security Problem

In our scheme, we consider the possible security problems from the following three points. One is the forgery of the signature, the second is the denial of the signature, and the third is the update of the group secret key. If an adversary forges the signature of a legitimate user in the system in order to deceive the cloud storage server to store the data on the server, the signature of legitimate users will be threatened. In signer identity tracking, the identity of signer is tracked by tracing the public key of the signer. If the signer wants to deny his signature, it will interfere with the tracking results and cause controversy. In view of the above problems, we need to ensure that the signature is non-forgery and non-repudiation. In the group key update phase, if an adversary can successfully update the group key, the adversary can masquerade the legitimate user to cheat. So we must also ensure that only the legitimate users in the system can update the group key.

4. Description of Proposed Scheme

In this section, we introduce our proposed scheme in detail. Our Scheme includes eight algorithms which are Setup Algorithm, Users Join Algorithm, Group Key Generation Algorithm, Signature Generation Algorithm, Upload Algorithm, Public Integrity Checking Algorithm, Trace Algorithm and Revocation Algorithm.

(1) Setup Algorithm: This algorithm is designed to generate system parameters. The group manager generates two order q cyclic multiplicative groups G_1 and G_T, which satisfies a

bilinear map $e: G_1 \times G_1 \to G_T$. g_1 is a generator of group G_1. Firstly, the group manager randomly chooses $ik \in Z_q^*$ as issuing key and calculates $P = g_1^{ik}$. Then the group manager randomly chooses $\varepsilon \in Z_q^*$. Finally two hash functions are defined, $H_1: \{0,1\}^* \to Z_q^*$ and $H_2: \{0,1\}^* \to G_1$ respectively. Let public parameter be $param = (G_1, G_T, H_1, H_2, g_1, P, \varepsilon)$.

(2) Users Join Algorithm: This algorithm is designed to generate user's public key and private key.

Let the group manager's public and private key be $(gmpk, gmsk)$. A user with identity ID_k encrypts his identity by the group manager's public key $gmpk$ and sends the encrypted identity to the group manager. The group manager decrypts the encrypted identity of the user with his private key $gmsk$ to obtain the user identity ID_k. The group manager randomly chooses $x_k \in Z_q^*$, calculates:

$$A_k = x_k + ik \cdot H_1(ID_k) \tag{3}$$

$$R_k = g_1^{x_k} \tag{4}$$

and returns A_k and R_k to the user.

After receives A_k and R_k, the user firstly verifies whether the equation $g_1^{A_k} = R_k \cdot P^{H_1(ID_k)}$ holds or not. If the equation holds, it means that A_k is valid. Then the user chooses $y_k \in Z_q^*$, and calculates:

$$sk_k = y_k + A_k \tag{5}$$

$$pk_k = g_1^{y_k} \tag{6}$$

as the user's private key and the public key respectively. Users who generate public and private keys based on the users join algorithm become legitimate users.

(3) Group Key Generation Algorithm: This algorithm is designed to generate group secret key. Group manager firstly calculates:

$$V_k = H_1(A_k) \tag{7}$$

and creates the register table which includes ID_k and V_k, saving the table locally. Then group manager creates a polynomial function $f(x) = \prod_{i=1}^{n}(x - V_i) = \sum_{i=0}^{n} a_i x^i \pmod q$, and defines a exponential function $\{W_0, W_1, \cdots, W_n\} = \{\varepsilon^{a_0}, \varepsilon^{a_1}, \cdots, \varepsilon^{a_n}\}$. Finally, the group manager randomly selects $K_g \in Z_p^*$ as the group secret key and calculates

$$\rho = g_1^{K_g} \tag{8}$$

$$EK = \{K_g \cdot W_0, W_1, \cdots, W_n\} \tag{9}$$

and publishes the group public key ρ and EK.

(4) Signature Generation Algorithm: This algorithm is designed to generate signature and the secret sharing. This algorithm consists of two phases which are signature generation and secret sharing generation.

The phase of signature generation is described in below. When the user with identify ID_k wants to upload file data M to the cloud, divides the file data into m blocks which are denoted as m_y, $1 \le y \le m$, and calculates:

$$V_k = H_1(A_k) \tag{10}$$

$$K_g \cdot W_0 \cdot \prod_{i=1}^{n} W_i^{V_k{}^i} = K_g \cdot \varepsilon^{f(V_k)} = K_g \tag{11}$$

$$\theta = (H_2(M))^{K_g} \tag{12}$$

Then the user randomly chooses $u \in G_1$ and calculates:

$$\delta_y = H_2(filename \| y) \cdot u^{m_y} \tag{13}$$

$$\sigma_y = \delta_y^{sk_k} \tag{14}$$

$$c_y = H_2(\sigma_y) \cdot \delta_y \tag{15}$$

where $1 \le y \le m$. Finally, the signature of the file data is completed.

The phase of secret sharing generation is described in below. Define the number of registered users in the system as n. Firstly the user randomly selects $\alpha_j \in Z_q^*$, $0 \le j \le t-1, 1 < t < n$ and constructs a polynomial function:

$$p(z) = \alpha_0 + \alpha_1 z + \cdots + \alpha_{t-1} z^{t-1} \tag{16}$$

where $1 < t < n$. The user keeps the polynomial secretly. Then the user calculates:

$$\tau_j = g_1^{\alpha_j},\ 0 \le j \le t-1 \tag{17}$$

$$\chi_i = \prod_{j=0}^{t-1} \tau_j{}^{i^j},\ 1 \le i \le n \tag{18}$$

$$\eta_i = pk_i^{p(i)},\ 1 \le i \le n \tag{19}$$

$$\gamma = g_1^{\alpha_0} \cdot pk_k \tag{20}$$

Finally, the user calculates:

$$s = H_2(\chi_1, \chi_2, \cdots, \chi_n, \eta_1, \eta_2, \cdots, \eta_n) \tag{21}$$

and sets $share = (\tau_0, \tau_1, \cdots, \tau_{t-1}, \eta_1, \eta_2, \cdots, \eta_n, s)$.

(5) Upload Algorithm: This algorithm is designed to upload user data and the metadata information of user data. The user uploads

$$\left\{ filename \| m, u, \theta, \{\sigma_y\}_{1 \le y \le m}, \{c_y\}_{1 \le y \le m}, \gamma, share, \{m_y\}_{1 \le y \le m} \right\}$$

to the cloud storage server. After receiving the data sent by the user, the cloud storage server firstly verifies the legality of the user's identity by verifying whether the equation $e(\theta, g_1) = e(H_2(M), \rho)$ holds. If the equation holds, it means that the user is a legitimate user. Otherwise, the cloud storage server refuses to serve the user. After ensuring the legality of the user's identity, the cloud storage server takes over the data and stores them to the sever.

(6) Public Integrity Checking Algorithm: This algorithm is designed to check data integrity. This algorithm consists of three phases: challenge generation phase, proof generation phase and proof verification phase.

In the challenge generation phase, mainly run by TPA. TPA picks a set L with l elements, where $L \subseteq [1,n]$, and chooses a random element $v_y \in Z_q^*$ for each $y \in L$. TPA generates a challenge $chal = filename \| \{(y, v_y)\}_{y \in L}$ and sends it to the cloud storage server.

In the proof generation phase, mainly run by cloud storage server. After receiving the challenge message $chal$ from the TPA, the cloud storage server firstly finds the corresponding file block $\{m_y\}_{y \in L}$ based on the file name $filename$ and index y, then calculates:

$$\mu = \sum_{y \in L} m_y v_y \tag{22}$$

$$T = \prod_{y \in L} {c_y}^{v_y} H_2(\sigma_y)^{v_y} \tag{23}$$

and sends (u, μ, T) to TPA.

In the proof verification phase, mainly run by TPA. After receives the proof from the cloud storage server, TPA checks whether the Eq.(24) holds or not.

$$T = u^{\mu} \prod_{y \in L} H_2(filename \| y)^{v_y} \tag{24}$$

If the equation holds, the TPA accepts the proof and returns "1" to user. Otherwise the proof is invalid, and the TPA returns "0" to user.

(7) Trace Algorithm: This algorithm is designed to track the identity of the signer. Firstly these valid users get the signer's secret sharing $shar$ and γ from the cloud. With $share = (\tau_0, \tau_1, \cdots, \tau_{t-1}, \eta_1, \eta_2, \cdots, \eta_n, s)$, any valid user calculates:

$$\chi_i = \prod_{j=0}^{t-1} {\tau_j}^{i^j}, \ 1 \le i \le n \tag{25}$$

$$s^* = H_2(\chi_1, \chi_2, \cdots, \chi_n, \eta_1, \eta_2, \cdots, \eta_n) \tag{26}$$

If $s^* = s$, t valid users can track the identity of the signer as follows. Each of the t valid users calculates:

$$\xi_i = \eta_i^{sk_i^{-1}} \tag{27}$$

$$\lambda_i = \prod_{j=1,2,\cdots,t, j \ne i} \frac{j}{j-i} \tag{28}$$

where $1 \le i \le t$ by using their private keys. Then the public key pk_k of the signer can be obtained by calculating $\gamma \cdot \left(\prod_{i=1}^{t} \xi_i^{\lambda_i} \right)^{-1}$.

(8) Revocation Algorithm: This algorithm is designed to revoke user and update the group key. After tracking to the user who needs to be revoked according to the tracking algorithm, the group manager queries the register table to find the registration information of the user who needs to be revoked. Group manager updates polynomial function and exponential function as follows:

$$f'(x) = f(x)/(x - V_k) = \sum_{i=0}^{n-1} a_i' x^i \pmod q \tag{29}$$

$$\{W_0', W_1', \cdots, W_{n-1}'\} = \{\varepsilon^{a_0'}, \varepsilon^{a_1'}, \cdots, \varepsilon^{a_{n-1}'}\} \tag{30}$$

Finally, the group manager updates the group secret key:

$$K_g = K_g' \tag{31}$$

and sets $EK' = \{K_g' \cdot W_0', W_1', \cdots, W_n'\}$.

4. Security Analysis

In this section we prove our scheme is correctness, traceability, full anonymity, unforgeability, non-repudiation and can safely update group key.

5.1 Correctness

In Users Join Algorithm, it is necessary to verify the validity of A_k by calculating $g_1^{A_k} = R_k \cdot P^{H_1(ID_k)}$ after the user receives A_k and R_k from the group manager. The correctness of the above equation can be proved as follows:

$$g_1^{A_k} = g_1^{x_k + ik \cdot H_1(ID_k)} = g_1^{x_k} \cdot g_1^{ik \cdot H_1(ID_k)} = R_k \cdot P^{H_1(ID_k)} \tag{32}$$

In Upload Algorithm, due to only legitimate user can enjoy the storage service, the cloud storage server must verify the identity of the user before storing data. Because only legitimate user has K_g, the cloud storage server can verify the legitimacy of the user's identity by calculating $e(\theta, g_1) = e(H_2(M), \rho)$.The correctness of the above equation can be proved as follows:

$$e(\theta, g_1) = e((H_2(M))^{K_g}, g_1) = e(H_2(M), g_1^{K_g}) = e(H_2(M), \rho) \tag{33}$$

In the proof verification phase of Public Integrity Checking Algorithm, TPA verifies data integrity by calculating $T = u^{\mu} \prod_{y \in L} H_2(filename \| y)^{v_y}$. The correctness of the above equation can be proved as follows:

$$\begin{aligned} T &= \prod_{y \in L} c_y^{v_y} H_2(\sigma_y)^{-v_y} \\ &= \prod_{y \in L} (H_2(\sigma_y) \cdot \delta_y)^{v_y} H_2(\sigma_y)^{-v_y} \\ &= \prod_{y \in L} \delta_y^{v_y} \\ &= \prod_{y \in L} (H_2(filename \| y) \cdot u^{m_y})^{v_y} \\ &= u^{\sum_{y \in L} m_y v_y} \cdot \prod_{y \in L} H_2(filename \| y)^{v_y} \\ &= u^{\mu} \cdot \prod_{y \in L} H_2(filename \| y)^{v_y} \end{aligned} \tag{34}$$

In Trace Algorithm, when t legitimate users track the identity of the signer, the public key pk_k of the signer is obtained by calculating $\gamma \cdot \left(\prod_{i=1}^{t} \xi_i^{\lambda_i} \right)^{-1}$. The proof of the correctness of the equation $pk_k = \gamma \cdot \left(\prod_{i=1}^{t} \xi_i^{\lambda_i} \right)^{-1}$ is as follows:

$$\begin{aligned} \gamma \cdot \left(\prod_{i=1}^{t} \xi_i^{\lambda_i} \right)^{-1} &= \gamma \cdot \left(\prod_{i=1}^{t} \left(\eta_i^{sk_i^{-1}} \right)^{\lambda_i} \right)^{-1} \\ &= \gamma \cdot \left(\prod_{i=1}^{t} \left(pk_i^{p(i)} \right)^{sk_i^{-1} \cdot \lambda_i} \right)^{-1} \\ &= \gamma \cdot \left(\prod_{i=1}^{t} \left(g_1^{sk_i} \right)^{p(i) \cdot sk_i^{-1} \cdot \lambda_i} \right)^{-1} \end{aligned} \tag{35}$$

$$
\begin{aligned}
&= \gamma \cdot \left(\prod_{i=1}^{t} g_1^{p(i) \cdot \lambda_i} \right)^{-1} \\
&= \gamma \cdot \left(\prod_{i=1}^{t} g_1^{p(i) \cdot \prod_{j=1,2,\cdots,t, j \neq i} \frac{j}{j-i}} \right)^{-1} \\
&= \gamma \cdot g_1^{-\sum_{i=1}^{t} p(i) \cdot \prod_{j=1,2,\cdots,t, j \neq i} \frac{j}{j-i}} \\
&= g_1^{\alpha_0} \cdot pk_k \cdot g_1^{-\alpha_0} \\
&= pk_k
\end{aligned}
$$

5.2 Traceability

Our scheme uses the secret sharing scheme to track the identity of the signer. Only no less than t legitimate users can track the identity of the signer. The signer generates a secret α_0 during the signature process, so these users who do the tracking can reconstructs the secret α_0 during the track process by using their private keys. Thereby they obtains the signer's public key, and tracks the identity of the signer.

Due to the traceability of the (*t*, *n*) threshold, any subset with more than t legitimate users can reveal the identity of the signer. A subset of users who do not meet the requirements cannot reveal the identity of the signer. Detailed proof can refer to [28], [29].

5.3 Full anonymity

The privacy of the user's identity can be ensured during the data upload phase and the data integrity verification phase. In the data upload phase, because only legitimate users can store data in the cloud storage server, the cloud storage server only needs to verify the legitimacy of the user's identity, and does not know the identity of the user. During the integrity verification phase, the verifier does not know the identity of the user and can still verify the integrity of the data. Therefore, full anonymity can be achieved.

5.4 Unforgeability

The unforgeability of signatures is mainly described in two parts, that is unforgeability of signature θ and $\{\sigma_y\}_{1<y<m}$.

(1) Unforgeability of signature θ: If an adversary tries to forge a signature θ^* and uploads data M^* to the cloud server. According to the CDH problem, knowing g_1, $\rho = g_1^{K_g} \in G_1$, $H_2(M^*) \in G_1$, it is difficult to calculate $\theta' = (H_2(M^*))^{K_g}$ without knowing k_g, so the adversary forges a signature $\theta^* \neq \theta'$. In the upload stage, the user's identity needs to be verified by determining whether the equation $e(\theta^*, g_1) = e(H_2(M^*), \rho)$ holds or not . Only when the above equation holds, can the cloud server accept the user's data.

$$
e(H_2(M^*), \rho) = e(H_2(M^*), g_1^{k_g}) = e((H_2(M^*))^{k_g}, g_1) = e(\theta', g_1) \neq e(\theta^*, g_1) \tag{36}
$$

It can be seen from the Eq.(36) that the above equation cannot be verified, so the cloud storage server will reject the data of the adversary.

(2) Unforgeability of signature $\{\sigma_y\}_{1<y<m}$: When the CDH assumption holds, the signature $\{\sigma_y\}_{1<y<m}$ can only be generated by the signer himself. Suppose an adversary wants to forge the signature of user ID^* about data $\{m^*_y\}_{1<y<m}$ in the system. He calculates $\delta^*_y = H_2(filename\|y)\cdot u^{m^*_y}$. According to the CDH assumption, knowing g_1, δ_y, pk^*, it is difficult to calculate $\sigma'_y = \delta_y^{sk^*} \in G_1$. Therefore, the forged signature by the adversary satisfied $\sigma^*_y = \sigma'_y$ is equivalent to solving the CDH problem. So only the signer with the private key can generate valid signature.

5.5 Non-repudiation

If a user in the system has illegal behavior, the user can be tracked by the tracking algorithm, at this time the identity of the user will no longer be kept confidential. In the signature algorithm, the user signs the data with his own private key. Once the user's identity is disclosed, the user's signature can be verified with the user's public key by verifying the Eq.(37):

$$e(\sigma_y, g_1) \stackrel{?}{=} e\left(H_2(filename\|y)\cdot u^{m_y}, pk\right) \tag{37}$$

Because the signature σ_y can not be forged, once the verification is passed, it means the signature belongs to the user, and the user can not deny it.

The accuracy of the verification is as follows:

$$\begin{aligned} e(\sigma_y, g_1) &= e\left(\delta_y^{sk}, g_1\right) \\ &= e\left(\left(H_2(filename\|y)\cdot u^{m_y}\right)^{sk}, g_1\right) \\ &= e\left(H_2(filename\|y)\cdot u^{m_y}, g_1^{sk}\right) \\ &= e\left(H_2(filename\|y)\cdot u^{m_y}, pk\right) \end{aligned} \tag{38}$$

5.6 Dynamic user revocation

Under the DL problem, only non-revoked users can update the group secret key. In the revocation algorithm, the group manager updates polynomial function $f'(x) = f(x)/(x-V_k) = \sum_{i=0}^{n-1} a_i' x^i \pmod q$ and exponential function $\{W_0', W_1', \cdots, W_{n-1}'\} = \{\varepsilon^{a_0'}, \varepsilon^{a_1'}, \cdots, \varepsilon^{a_{n-1}'}\}$ and updates the group secret key to K_g'. For a non-revoked user $ID_{\tilde{k}}$, he can calculate $V_{\tilde{k}} = H_1(A_{\tilde{k}})$, $V_{\tilde{k}} \in \{V_i\}_{1\le i\le n-1}$. Let $x = V_{\tilde{k}}$, $f'(V_{\tilde{k}}) = \prod_{i=1}^{n-1}(V_{\tilde{k}} - V_i) = 0$. This polynomial can only output 0 when $1 \le \tilde{k} \le n-1$. So only non-revoked users can calculate $K_g' \cdot W_0' \cdot \prod_{i=1}^{n-1} W_i^{V_{\tilde{k}}^i} = K_g' \cdot \varepsilon^{f'(V_{\tilde{k}})} = K_g'$. If an adversary wants to calculate a group secret key K_g', given g_1 and $\rho = g_1^{K_g} \in G_1$, he should deal with DL problem, which is impossible.

6 Performance Analysis

In this section, we evaluate the performance of our scheme by calculating the computation cost and communication cost. By comparing our scheme with Huang [14] and Hu [28], we can analyze the performance of our scheme more clearly. The relevant symbols are shown in **Table 1**.

Table 1. Notation

Notation	Descriptions
n	Number of legitimate users
t	Number of users to track
m	Number of blocks per file
l	Number of blocks challenged
l_{name}	Length of filename
l_s	Length of challenge index
l_q	Length of the element on Z_q^*
l_{G_1}	Length of the element on G_1
H_1	Hash function in Z_q^*
H_2	Hash function in G_1
Exp_{G_1}, $Exp_{Z_q^*}$	Exponentiation in G_1, Z_q^*
Mul_{G_1}, $Mul_{Z_q^*}$, Mul_{G_T}	Multiplication in G_1, Z_q^*, G_T
Pair	Paring operation $e: G_1 \times G_1 \to G_T$

6.1 Computation Cost

We mainly consider computation cost in two aspects: the efficiency of signature and the efficiency of public integrity checking.

(1) Signature efficiency: In the signature generation process, the total computation cost is

$$H_1 + (2m+2)H_2 + (\mathrm{nt}+2n)Exp_{Z_q^*} + nMul_{Z_q^*} + (nt+n+t+2m+2)Exp_{G_1} + (nt-n+2m+1)Mul_{G_1}.$$

The signature algorithm consists of two phases: the signature generation phase and the secret sharing generation phase. In the signature generation phase, the user needs to calculate $\{V_k, K_g, \theta, \{\delta_y\}_{1\le y\le m}\{\sigma_y\}_{1\le y\le m}, \{c_y\}_{1\le y\le m}\}$. As shown in **Table 2**, in this phase, computation cost is

$$H_1 + (2m+1)H_2 + 2nExp_{Z_q^*} + nMul_{Z_q^*} + (2m+1)Exp_{G_1} + 2mMul_{G_1}.$$

In the secret sharing generation phase, the user needs to calculate $\{\{\tau_j\}_{0\le j\le t}, \{\chi_i\}_{1\le i\le n}, \{\eta_i\}_{1\le i\le n}, \gamma, s\}$ to obtain secret sharing $share$. As shown in **Table 2**, in this phase, computation cost is

$$H_2 + ntExp_{Z_q^*} + (nt+n+t+1)Exp_{G_1} + (nt-n+1)Mul_{G_1}.$$

In addition, we display the computation cost of the signature for Huang's and Hu's schemes in **Table 2**.

Table 2. Computation cost of signature

Scheme		Computation cost
Hu's scheme	Signature generation	$(m+1)H_1+(2m+1)H_2+3Pair+(4m+n+1)Mul_{Z_q^*}+2nExp_{Z_q^*}+(5m+6)Exp_{G_1}+(3m+4)Mul_{G_1}+3mExp_{G_T}+2mMul_{G_T}$
Huang's scheme	Secret sharing generation	$H_2+ntMul_{Z_q^*}+(nt+n+t)Exp_{G_1}+(nt-n)Mul_{G_1}$
	Signature generation	$mH_2+(2nm+m)Exp_{G_1}+2mMul_{G_1}$
Ours	Secret sharing generation	$H_2+ntExp_{Z_q^*}+(nt+n+t+1)Exp_{G_1}+(nt-n+1)Mul_{G_1}$
	Signature generation	$H_1+(2m+1)H_2+2nExp_{Z_q^*}+nMul_{Z_q^*}+(2m+1)Exp_{G_1}+2mMul_{G_1}$

Table 3. Computation cost of public integrity checking

Scheme		Computation cost
Hu's scheme	Cloud	$lH_1+lH_2+2Pair+lMul_{Z_q^*}+9lExp_{G_1}+(5l-1)Mul_{G_1}+4lExp_{G_T}+(3l+1)Mul_{G_T}$
	TPA	$lH_2+(l+1)Exp_{G_1}+lMul_{G_1}$
Huang's scheme	Cloud	$(nl+1)Exp_{G_1}+(nl-n)Mul_{G_1}+lMul_{Z_q^*}$
	TPA	$nH_1+lH_2+(n+2)Pair+(l+1)Exp_{G_1}+lMul_{G_1}+nMul_{G_T}$
Ours	Cloud	$lH_2+lMul_{Z_q^*}+2lExp_{G_1}+(2l-1)Mul_{G_1}$
	TPA	$lH_2+(l+1)Exp_{G_1}+lMul_{G_1}$

(2) Public integrity checking efficiency: In public integrity checking process, the total computation cost is $2lH_2+lMul_{Z_q^*}+(3l+1)Exp_{G_1}+(3l-1)Mul_{G_1}$. The public integrity checking algorithm consists of three phases: the challenge generation phase, the proof generation phase, and the proof verification phase. Since there is no computation cost in the challenge generation phase, the computation cost of public integrity checking algorithm is mainly generated by the proof generation phase and the proof verification phase. In the proof generation phase, the cloud storage server needs to calculate $\{\mu,T\}$. The computational cost is $lH_2+lMul_{Z_q^*}+2lExp_{G_1}+(2l-1)Mul_{G_1}$, as shown in **Table 3**. In the proof verification phase, TPA performs verification. Its computational cost is $lH_2+(l+1)Exp_{G_1}+lMul_{G_1}$ as shown in **Table 3**. In addition, we display the computation cost of public integrity checking for

Huang's and Hu's schemes in **Table 3**.

6.2 Communication Cost

We compared the communication overhead of our scheme with Huang's scheme [14] and Hu's scheme [28]. During the challenge generation process of our scheme, TPA sends challenge $chal = filename \| \{(y, v_y)\}_{y \in L}$ to the cloud storage server and its communication cost is $l \cdot l_q + l \cdot l_s + l_{name}$. In the proof generation stage, the cloud storage server sends the proof $\{u, \mu, T\}$ to the TPA, and its communication overhead is $l_q + 2l_{G_1}$. Therefore, during public integrity checking process, the total communication cost of our scheme is $(l+1)l_q + l \cdot l_s + l_{name} + 2l_{G_1}$ as shown in **Table 4**. In addition, we display the communication cost of Huang's and Hu's schemes in **Table 4**.

Table 4. Communication cost

Scheme	Communication cost
Hu	$(1+l)l_q + l \cdot l_s + l_{name} + 2l_{G_1}$
Huang	$(2l+1)l_q + l \cdot l_s + (n+1)l_{G_1}$
Ours	$(l+1)l_q + l \cdot l_s + l_{name} + 2l_{G_1}$

6.3 Experimental Results

In this paper, we analyze the efficiency of our scheme and compare it with Huang's scheme [14] and Hu's scheme [28]. Our experiment was carried out under the windows platform and used java programming language with Java Paring Based Cryptography(JPBC). We set base field size to be 160bits and the size of an element in Z_q^* to be 160bits.

The efficiency of signature generation can be shown in **Fig. 2**. We set $m = 250$ and $t = 5$. When the number of legitimate users in the system increases from 50 to 150, the signature time in our scheme and Hu's scheme [28] increases slowly with the increase of the number of users in the system, but the signature time of Hu's scheme [28] is slightly more than that of our scheme. However, the signature time of Huang's scheme [14] is greatly affected by the number of users in the system, and the signature time increases linearly with the increase of the number of users.

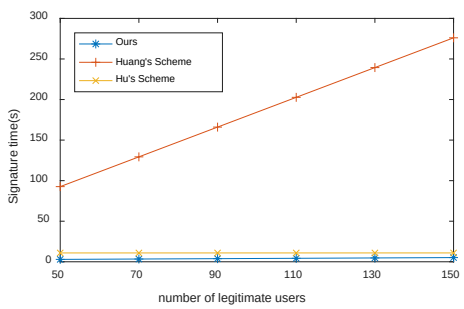

Fig. 2. Comparison of signature time

The efficiency of public integrity checking can be shown in **Fig. 3** (a) and **Fig. 3** (b). In order to achieve detection rate of 99%, we also set $m = 10000$, $l = 460$ like Hu's scheme [28]. As shown in **Fig. 3** (a), when the number of blocks to be detected is $l = 460$, computation time of cloud storage server in our scheme and Hu's scheme [28] is constant and independent of the number of users. And our scheme take less time than Hu's scheme [28], because Hu's scheme [28] needs to calculate some relevant values of zero-knowledge proof to generate proof. However, computation cost of cloud storage server in Huang's scheme [14] is greatly affected by the number of users in the system, and the signature time increases linearly with the increase of the number of users. As shown in **Fig. 3** (b), computation time of TPA in our scheme and Hu's scheme [28] is also constant and independent of the number of users. And our scheme takes less time than Hu's scheme [28]. However, computation cost of TPA in Huang's scheme [14] increases linearly with the increase of the number of users.

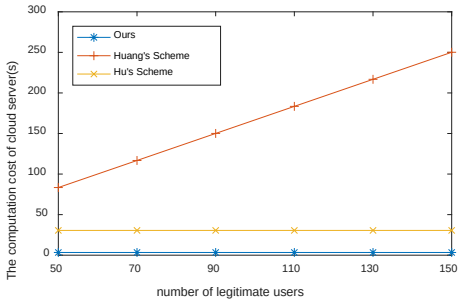

Fig. 3. (a) Comparison of computation cost of cloud storage server

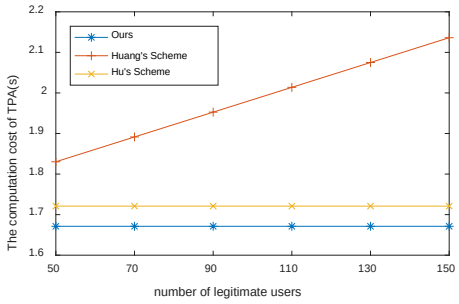

Fig. 3. (b) Comparison of computation cost of TPA

In the process of public integrity checking, the communication cost is shown in **Fig. 4**. We set $l_s = 32$, $l_{name} = 128$. The communication overhead in Huang's scheme [14] increases linearly with the increase of the number of users. The communication overhead in our scheme and Hu's scheme [28] has nothing to do with the number of users, and our scheme has less communication overhead than Hu's scheme [28]. Obviously our scheme is more effective than Huang's and Hu's schemes.

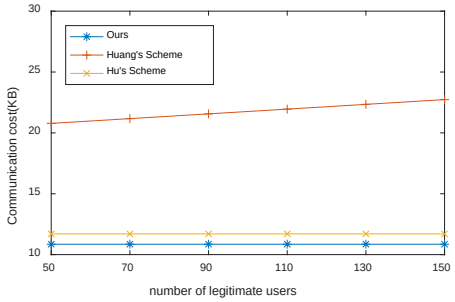

Fig. 4. Comparison of communication cost

7. Conclusion

In this paper, we propose a traceable dynamic public auditing with identity privacy preserving for cloud storage, which supports dynamic user operation. In addition, our scheme realizes full anonymity and traceability. In the proposed scheme, a certain number of legitimate users can work together to trace the identity of the signer, but a single user, including a group manager, is unable to trace the identity. The security and efficiency analysis shows that our scheme is secure and efficient.

References

[1] W. Shen, J. Qin, J. Yu, R. Hao, J. Hu, "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 331-346, 2018. Article (CrossRef Link).

[2] J. Shen, J. Shen, X. Chen, X. Huang, W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402-2415, 2017. Article (CrossRef Link).

[3] T. Jiang, X. Chen, and J. Ma, "Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2363-2373, 2016. Article (CrossRef Link).

[4] Y. Zhang, R. Deng, D. Zheng, J. Li, P Wu, and J. Cao, "Efficient and Robust Certificateless Signature for Data Crowdsensing in Cloud-assisted Industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 5099-5108, 2019. Article (CrossRef Link).

[5] B. Kuang, A. Fu, S. Yu, G. Yang, M. Su, Y. Zhang, "ESDRA: An Efficient and Secure Distributed Remote Attestation Scheme for IoT Swarms," *IEEE Internet of Things Journal*, vol. 6, pp. 8372-8383, 2019. Article (CrossRef Link).

[6] Y. Yu, M. 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 Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 767-778, 2016. Article (CrossRef Link)

[7] Y. Zhang, J. Yu, R. Hao, C. Wang, and K. Ren, "Enabling efficient user revocation in identity-based cloud storage auditing for shared big data," *IEEE Transactions on Dependable and Secure Computing*, pp. 1-1, 2018. Article (CrossRef Link).

[8] Q. Wang, C. Wang, K. Ren, W. Lou, J. Lin, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE transactions on parallel and distributed systems*, vol. 22, no. 5, pp. 847-859, 2011. Article (CrossRef Link).

[9] B. Wang, B. Li, and H. Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud," *IEEE Transactions on Services Computing*, vol. 8, no. 1, pp. 92-106, 2015. Article (CrossRef Link).

[10] Y. Luo, M. Xu, S. Fu, D. Wang, and J. Deng, "Efficient Integrity Auditing for Shared Data in the Cloud with Secure User Revocation," in *Proc. of IEEE Trustcom/BigDataSE/ISPA*, pp. 434-442, 2015. Article (CrossRef Link).

[11] A. Fu, S. Yu, Y. Zhang, H. Wang, C. Huang, "NPP: A New Privacy-Aware Public Auditing Scheme for Cloud Data Sharing with Group Users," *IEEE Transactions on Big Data*, pp. 1-1, 2017. Article (CrossRef Link).

[12] B. Wang, B. Li, and H. Li, "Oruta: privacy-preserving public auditing for shared data in the cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 43-56, January, 2014. Article (CrossRef Link).

[13] L. Huang, G. Zhang, A. Fu, "Privacy-Preserving Public Auditing for Non-Manager Group," in *Proc. of IEEE ICC*, pp. 1-6, 2017. Article (CrossRef Link).

[14] L. Huang, G. Zhang, A. Fu, "Privacy-Preserving Public Auditing for Non-Manager Group," *Wireless Personal Communication*, vol. 100, no. 4, pp. 1277-1294, 2018. Article (CrossRef Link).
[15] J. Xiong, Y. Zhang, X. Li, M. Lin, Z. Yao, "RSE-PoW: A role symmetric encryption PoW scheme with authorized deduplication for multimedia data," *Mobile Networks and Applications*, vol. 23, no. 3, pp. 650-663, 2018. Article (CrossRef Link).
[16] Y. Zhang, D. Zheng, R. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2130-2145, 2018. Article (CrossRef Link).
[17] D. Wang, D. He, P. Wang, and C. Chu, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428-442, 2015. Article (CrossRef Link).
[18] D. Wang, P. Wang, "On the Anonymity of Two-Factor Authentication Schemes for Wireless Sensor Networks: Attacks, Principle and Solutions," *Computer Networks*, vol. 73, no. 14, pp. 41–57, 2014. Article (CrossRef Link).
[19] Y. Zhang, R. Deng, X. Liu, and D. Zheng, "Outsourcing service fair payment based on blockchain and its applications in cloud computing," *IEEE Transactions on Services Computing*, pp. 1-1, 2018. Article (CrossRef Link).
[20] Y. Zhang, R. Deng, X. Liu, and D. Zheng, "Blockchain based Efficient and Robust Fair Payment for Outsourcing Services in Cloud Computing," *Information Sciences*, vol. 462, pp. 262-277, 2018. Article (CrossRef Link).
[21] Y. Zhang, R. Deng, J. Shu, K. Yang, D. Zheng, "TKSE: Trustworthy Keyword Search over Encrypted Data with Two-side Verifiability via Blockchain," *IEEE Access*, vol. 6, pp. 31077-31087, 2018. Article (CrossRef Link).
[22] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1717-1726, August, 2015. Article (CrossRef Link).
[23] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546-2559, 2016. Article (CrossRef Link).
[24] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *Journal of Internet Technology*, vol. 16, no. 2, pp. 317-323, 2015. Article (CrossRef Link).
[25] G. Ateniese et al., "Provable data possession at untrusted stores," in *Proc. of ACM Conference on Computer & Communications Security*, pp. 598-609, 2007. Article (CrossRef Link).
[26] H. Shacham, and B. Waters, "Compact proofs of retrievability," in *Proc. of Cryptology-ASIACRYPT 2008*, pp. 90-107, 2008. Article (CrossRef Link).
[27] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297-319, 2004. Article (CrossRef Link).
[28] A. Hu, R. Jiang, B. Bhargava, "Identity-Preserving Public Integrity Checking with Dynamic Groups for Cloud Storage," *IEEE Transactions on Services Computing*, pp. 1-1, July, 2018. Article (CrossRef Link).
[29] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attack," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281-308, 1998. Article (CrossRef Link).
[30] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42-61, 2017. Article (CrossRef Link).
[31] Y. Zhang, A. Wu, and D. Zheng, "Efficient and privacy-aware attribute-based data sharing in mobile cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1039-1048, 2018. Article (CrossRef Link).
[32] X. Li, H. Qian, J. Li, "Democratic group signatures with threshold traceability," *Journal of Shanghai Jiaotong University*, vol. 14, no. 1, pp. 98-101, 2009. Article (CrossRef Link).

[33] G. He, X. Li, Q. Li, D. Zheng, "Efficient democratic group signatures with threshold traceability," *Journal of Shanghai Jiaotong University*, vol. 16, no. 5, pp. 530-532, 2011. Article (CrossRef Link).
[34] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612-613, 1979. Article (CrossRef Link).
[35] Z. Zhu and R. Jiang, "A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 40-50, January, 2016. Article (CrossRef Link).

Yinghui Zhang is a professor of National Engineering Laboratory for Wireless Security (NELWS), Xi'an University of Posts & Telecommunications since 2018. He has published over 80 research articles in ACM ASIACCS, IEEE Transactions on Services Computing, Computer Networks, IEEE Internet of Things Journal, Computers & Security, IEEE Transactions on Industrial Informatics, IEEE Transactions on Dependable and Secure Computing, etc. His research interests include public key cryptography, cloud security and wireless network security.

Tiantian Zhang is currently pursuing her master degree at National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications. Her research interests include cloud storage security and data integrity auditing.

Rui Guo received the Ph.D. degree from Beijing University of Posts and Telecommunications, China, in 2014. He is currently a Lecturer with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications. His present research interests include attribute-based cryptograph, cloud computing, and blockchain technology.

Shengmin Xu received the B.Sc. degree in the School of Computing and Information Technology, University of Wollongong, Australia, in 2014 and Ph.D. degree in Cryptography from University of Wollongong, Australia, in 2018. He is currently a research fellow at School of Information System, Singapore Management University, Singapore. His research interests include cryptography and information security.

Dong Zheng received his Ph.D. degree in communication engineering from Xidian University, China, in 1999. He was a Professor at the School of Information Security Engineering, Shanghai Jiao Tong University. He is currently a Professor at National Engineering Laboratory for Wireless Security, Xi'an University of Posts & Telecommunications. He has published over 100 research articles including CT-RSA, IEEE Transactions on Industrial Electronics, Information Sciences, etc. His research interests include cloud computing security, public key cryptography.