ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20221216171505

Original scientific paper

Identity-Based Integrity Verification and Public Auditing Scheme in Cloud Storage System Against Malicious Auditors

Haojue ZHANG, Yilin YUAN*, Xianwei XIN, Yanbo QU

Abstract: The cloud storage system provides users with convenient storage services. However, users who use cloud storage services lose absolute control over the data once they upload files to the cloud service provider (CSP). The integrity of the cloud data has become a problem worth considering. In the realization of verification of cloud storage schemes, a third-party auditor (TPA), as a professional organization that provides users with public audit services, is an indispensable and important entity. However, the TPA is not entirely credible because the TPA may perform audits dishonestly out of laziness or selfishness. Based on the above considerations, we provide a scheme for solving the problem of cloud data integrity verification and the TPA dishonesty based on identity-based encryption (IBE). The proposed identity-based public auditing verification scheme can provide security against malicious auditors. In this scheme, the mathematical design is based on IBE, which frees the users from complicated certificate management. In addition, during the public auditing phase, adding a timestamp can effectively prevent the malicious TPA from working dishonestly. Moreover, the security analysis and performance evaluation of the scheme for the untrusted CSP and semi-trusted TPA shows effective results.

Keywords: cloud storage; identity-based encryption; integrity verification; public auditing

1 INTRODUCTION

With the rapid development of cloud computing, cloud storage service has become a widely known term. Cloud storage system provides users with large capacity, efficient access, flexibility, and storage services at a reasonable price, and users can choose different types of cloud storage services according to their own needs, including public cloud storage, private cloud storage, and hybrid cloud storage. In the public cloud storage, the CSP provides service for users through the internet or other access interface, and users can choose service mode (free service or paid service) according to their demand. There are many advantages of public cloud storage: users enjoy the service without high configuration hardware, reasonable bandwidth cost, simple installation, low cost, etc., which makes it popular. However, in the public cloud storage, due to the transparency of the CSP end to the user end (which means the user does not know any operation of the CSP), the integrity of the users' outsourced data becomes a problem worth considering. Thus, it is valuable to study the security and integrity of data when using public cloud storage services.

1.1 Motivation

Users using cloud storage services upload their data to the cloud server to save local storage space. Although the CSP provides users with large-capacity storage space, it is not always reliable. For example, for saving storage space, the CSP may discard the data that users do not use frequently; server-side downtime may also result in file loss, while the CSP chooses to conceal for the sake of maintaining its honour. Therefore, for knowing the data integrity, users need to regularly verify the data integrity on the CSP. Furthermore, verifying the data integrity is a crucial issue in securing cloud storage.

At times, the users cannot verify data integrity (for instance, the users have a business travel continuously in a place without network, or the user is imprisoned) for some reason; so, the users need to verify data integrity with the help of others. This "others" performs data integrity verification for the users under the authorization, but only if the user informs it of certain information related to the stored data. On the other hand, regularly verifying data integrity is a burdensome duty for the users with low-capacity and low-performance devices. Based on the above considerations, users can choose public audit, and further, the concept of third-party auditor (TPA) is proposed. The TPA is a professional organization that provides public audit services for users, and it is also an indispensable part of the cloud storage scheme that implements public integrity audits.

As it is known that the CSP is not reliable and the TPA is not entirely credible, there are many false practices that the TPA follows, which include reducing the number of audits due to laziness purpose, exploring users' sensitive data out of curiosity, and colluding with the CSP to deceive users. Therefore, when designing a cloud storage scheme that supports public audit, it is necessary to not only ensure the security of the scheme itself (against the untrusted CSP) but is also necessary to put forward countermeasures against malicious TPA (to prevent the malicious TPA from dishonestly performing audit work).

1.2 Contribution

Under the public cloud storage, we propose a public data integrity auditing scheme based on identity-based encryption (IBE). This paper aims to solve the remote data integrity checking and prevent the untrusted TPA from dishonestly performing audit work, thereby combating malicious auditors. The contributions are summarized as follows:

(1) The mathematical design of the scheme is based on IBE, freeing users from complicated certificate management. During the public auditing phase, the dishonest behavior of the malicious TPA can be effectively prevented by adding timestamps.

(2) For the treacherous CSP, we design a random oracle model based on the CDH problem, which proves the soundness of the scheme; and furthermore, since the TPA is semi-trusted, with the help of three attack models, the security of TPA and CSP is verified.

(3) A series of experimental evaluation validates the effectiveness of our proposed scheme.

1.3 Paper Organization

The rest of the paper is organized as follows. In section 2, we present the literature review. In section 3, the research methodology is given. The security analysis is elaborated in section 4. The result and conclusion are presented in section 5.

2 LIRERATURE REVIEW

To verify the integrity of remote data, provable data possession (PDP) [1] scheme is proposed. It is a probabilistic checking model, using RSA to design homomorphic verification tags (HVTs), so the calculation cost will increase with the amount of outsourced data, and cannot be applied to dynamic data. To realize the dynamics of the cloud data, more dynamic PDP schemes [2-4] are put forward. Since PDP is an efficient scheme to verify integrity, improved PDP schemes are raised [5-7]. Besides, another important model to implement data integrity verification is proof of retrievability (POR) [9]. The HVTs of the POR scheme adopt the bilinear map, introduce erasure code technology, and add a special data block named "sentinels", so POR is a retrievable model. Based on the POR scheme, Shacham [10] proposed an improved scheme. When the HVTs are built by pseudorandom functions (PRFs), the scheme can support private verification and is secure in the standard model; when the HVTs are generated by BLS signature, the scheme can support public verification and is secure in the random oracle model. And this scheme is the first to prove security against arbitrary adversaries. Furthermore, more interesting public verification schemes are designed [11-14].

The TPA, an important organization, is involved in the implementation of public verification and its emergence frees users from the huge and repeated audit work. The prerequisite for the security of most existing integrity audit schemes is to assume that the TPA is trusted, but its own trustworthiness is questionable. To prevent the malicious TPA from dishonestly performing audit work, Armknechtet [15] proposed the first scheme for verifying data integrity against malicious auditors, which generated verification information by bitcoin work certification mechanism. Following the work of Armknecht et al., Zhang [16] put forward a public integrity verification scheme, which not only supports certificate less public verification but also resists malicious auditors.

Most of the existing public auditing schemes are based on public key infrastructure (PKI), which distributes public and private keys for users. However, the scheme based on PKI has to face various issues related to certificate management, such as certificate storage, forwarding, verification, etc., and in reality, the efficiency of certificate management is low and cumbersome. To get rid of those problems, the public auditing scheme based on IBE, namely identity-based public audit (IBPA) [17-18] was proposed. Xue et al. [19] designed a public auditing scheme against malicious auditors with the help of the Bitcoin blockchain. In this scheme, it is stipulated that every time the TPA performs auditing work, it must generate a Nonce field according to the time stamp selected by the user, and record the Nonce field, time stamp and all related contents to a log file. Since log files need to be regularly uploaded to the blockchain to ensure that the content will not be tampered with and that the TPA cannot be denied, it will inevitably bring additional storage and computing overhead. Zhang et al. [20] designed a public audit scheme with the help of the Ethereum blockchain, assuming that TPA delays auditing, and users are not completely innocent. Subsequently, more valuable schemes [24-29] were proposed.

3 RESEARCH METHODOLOGY

3.1 Preliminaries

3.1.1 Bilinear Map

Let G_1, G_2 be the multiplicative cyclic group with the order p, g is the generator of G_1. A Bilinear Pairing $e: G_1 \times G_1 \to G_2$ satisfies the following properties:

a) Bilinearity: $\forall u, v \in G_1$ and $\forall a, b \in Z_P^*$, $e\left(u^a, v^b\right) = e(u, v)^{ab}$.

b) Non-degeneracy: $e(g, g) \neq 1$.

c) Computable: there is an efficient algorithm to calculate e.

3.1.2 Computational Diffie-Hellman (CDH) Problem

For unknown $a, b \in Z_p^*$, given g, g^a and g^b as input, output $g^{ab} \in G_1$. The CDH assumption in G_1 holds if it is computationally infeasible to solve the CDH problem in G_1.

3.2 System Architecture

The system architecture is shown in Fig. 1, and the system consists of four entities:

(1) Users: the users of the public cloud storage services, possess outsourced data that need to be uploaded to the CSP.

(2) Cloud Service Provider (CSP): an untrusted entity that can provide public cloud storage service.

(3) Private Key Generator (PKG): the trusted entity. The PKG generates the private key according to the user's identity.

(4) Third Party Auditor (TPA): the semi-trusted entity authorized by the user to perform public audit.

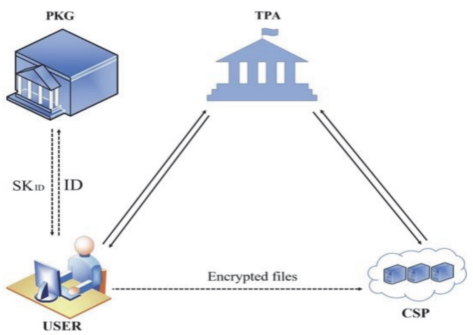


Figure 1 The system architecture

3.3 Algorithms

The proposed scheme consists of six algorithms: *Setup*, *Extract*, *TagGen*, *Challen*, *ProofGen*, *ProofVerify*, and the described are as follows:

- *Setup*: this algorithm is run by the PKG. It takes the security parameter *k* as input, and outputs system parameter (*pp*), the user's master public key (*mpk*), and master secret key (*msk*).

- *Extract*: this algorithm is run by the PKG. It takes the user's master public key (*mpk*), master secret key (*msk*), and user identity *ID* as input, and outputs the user's private key SK_{ID}.

- *TagGen*: this algorithm is run by the user. It takes the encrypted file *F*, the user *ID*, user's private key SK_{ID} as input, and outputs the tags set *T*.

- *Challen*: this algorithm is run by the TPA. The TPA generates the integrity challenge *chal* and sends it to the CSP.

- *ProofGen:* this algorithm is run by the CSP. The CSP generates the audit proof *P* according to the *chal*.

-*ProofVerify*: this algorithm is run by the TPA. It takes the integrity challenge *chal*, master public key *mpk*, and system parameter *pp* as input, and outputs the "1" or "0".

3.4 Security Model

We assume that the CSP is untrustworthy and TPA is semi-trustworthy. Here we propose the security models for the CSP and TPA respectively.

3.4.1 The Security Models for CSP

To formalize the security model, we put forward a game between challenger $\mathcal{C}$ and adversary $\mathcal{A}$. The user *ID* is regarded as challenger $\mathcal{C}$, and the untrusted CSP is regarded as adversary $\mathcal{A}$, and specific phases are given as follows:

(1) Setup phase:

The challenger $\mathcal{C}$ runs the *Setup* algorithm to obtain the system parameter *pp*, user's master public key (*mpk)*, master secret key (*msk)*, and then sends *pp* and *mpk* to the adversary $\mathcal{A}$.

(2) Query phase:

The adversary $\mathcal{A}$ makes two types of queries to the challenger $\mathcal{C}$ in this phase.

a) *Extractquery*. The adversary $\mathcal{A}$ queries the user's private key. The challenger $\mathcal{C}$ runs the *Extract* algorithm to get the user's private key SK_{ID} and return it to the adversary $\mathcal{A}$.

b) *TagGenquery*. The adversary $\mathcal{A}$ queries the tags set of the encrypted file *F*. The challenger *C* runs the *TagGen* algorithm to obtain the tags set and reply to the adversary $\mathcal{A}$.

(3) Challenge phase:

In this phase, the adversary $\mathcal{A}$ acts as the prover and the challenger $\mathcal{C}$ as the verifier. The challenger $\mathcal{C}$ sends an integrity challenge *chal* to the adversary $\mathcal{A}$ and requires the adversary $\mathcal{A}$ to provide an audit proof *P*.

(4) Forgery phase:

Upon received *chal*, the adversary $\mathcal{A}$ output and reply to the audit proof *P* to the challenger $\mathcal{C}$. If the audit proof *P* can pass the verification of the challenger $\mathcal{C}$ with non-negligible probability, we say the adversary $\mathcal{A}$ wins the game.

In the above security model, if the adversary $\mathcal{A}$ does not truly store all the blocks in the challenged set, it cannot generate the valid integrity proof and pass the $\mathcal{C}$'s verification.

Definition 1. The following condition must be true if the proposed scheme is secured; whenever the audit proof *P* generated by an adversary $\mathcal{A}$ passes the $\mathcal{C}$'s verification with non- negligible probability, there exists a knowledge extractor that can extract the challenged data blocks except negligible probability.

3.4.2 The Security Models for CSP/TPA

Here, we define three types of attacks according to reference [23].

Definition 2. (1) Replace attack. If an adversary has discarded (b_i, σ_i), for passing the audit, he will replace it with valid and uncorrupted pair $\left(b_k, \sigma_k\right)$.

(2) Forgery attack. An adversary may forge the audit proof to cheat the TPA.

(3) Replay attack. Without retrieving the real outsourced data, the adversary may generate audit proof with the help of previous proof in an attempt to pass the TPA's audit (the adversary may replay the previous integrity proof in an attempt to pass the TPA's audit).

In our proposal, suppose that the CSP may launch the above three attacks, and the TPA is honest but curious and may launch forgery attacks.

3.5 The Proposed Scheme

The proposed scheme includes six algorithms, which are introduced in detail as follows.

(1) Setup Algorithm:

a) the PKG chooses two multiplicative cyclic groups G_1 and G_2 of prime order p, and g is a generator of G_1. The PKG selects cryptographic hash function $H:\{0,1\}^* \rightarrow G_1$, the bilinear pairing map $e: G_1 \times G_1 \rightarrow G_2$ and the Pseudo-random function $f: Z_P^* \times Z_P^* \rightarrow Z_P^*$.

b) The PKG randomly selects some elements $\mu, u_1, u_{2,} ... u_n \in G_1$.

c) The PKG selects an element $x \in Z_P^*$ as master secret key and computes the master public key $mpk = g^x$.

d) The PKG sets the system parameter $pp = (G_1, G_2, g, p, H, e, \mu, u_1, u_{2,}, ..., u_n)$. Then the PKG publishes the *pp* and *mpk*, and keeps *msk* secret.

(2) Extract Algorithm:

According to the user's *ID*, the PKG picks $r_{ID} \in Z_P^*$ and computes $sk'_{ID} = g^{rID}$, $sk''_{ID} = g^x \cdot (\prod_{l=1}^{n} u_l^{ID})^{rID}$.

The user's private key is $sk_{ID} = \left(sk'_{ID}, sk''_{ID}\right) = \left(g^{rID}, g^x \cdot \prod_{l=1}^{n} u_l^{ID}\right)^{rID}$. The PKG sends the private key to the user *ID* through secure channel.

(3) TagGen Algorithm:

a) The sourced file of the user *ID* is F'. To generate the tags set, the user *ID* encrypts the sourced file and divides the encrypted file into n blocks $F = \{b_i\}_{1 \le i \le n}$, where $F = E_{key}(F')$, *key* is a secret key only visible to the user *ID*.

b) The user *ID* chooses a random element $\mu,\ rr \in Z_P^*$ and computes g^{rr}.

c) For block b_i, the user *ID* computes $T = \{\sigma_i\}_{1 \le i \le n}$, where $\sigma_i = g^x \cdot \left(\prod_{l=1}^{n} u_l^{ID}\right)^{r_{ID}} \cdot \left(H(name \,||\, ID_i)\,\mu^{b_i}\right)^{rr}$, $name \in Z_p^*$ is a random value selected as the file identifier, and $ID_i \in \{0,1\}^*$, $1 \le i \le n$ is the block identifier.

d) The user *ID* sends $\{F, T\}$ to the CSP and then deletes the local storage.

(4) Challenge Algorithm:

During public auditing process, the TPA sends the integrity challenge *chal* to the CSP. The detailed process is given as follows:

a) The TPA generates a timestamp t based on the current time.

b) Based on timestamp *t*, the TPA selects a random subset *I* with *ll*-elements, where $I \in [1, n]$.

c) The TPA generates $v_i \in Z_P^*$ for each $i \in I$.

d) The TPA sends the $chal = \{i,\ v_i\}_{1 \le i \le n}$ to the CSP.

(5) ProofGen Algorithm:

After receiving the *chal*, the CSP generates the audit proof.

a) The CSP calculates $\sigma = \prod_{i \in I} \sigma_i^{v_i}$, $\lambda = \sum_{i \in I} b_{ij} v_i$.

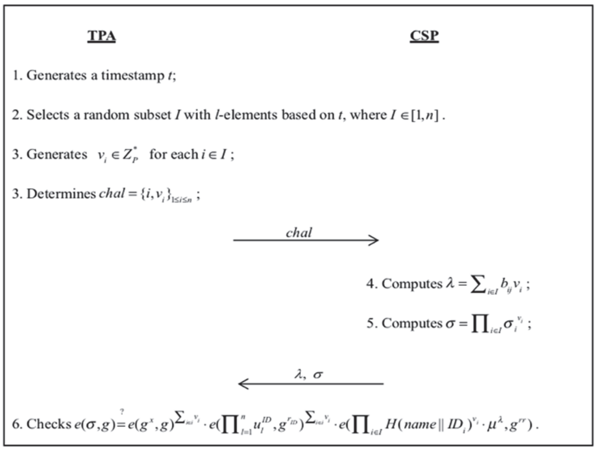
b) The CSP outputs the audit proof $P = \{\sigma, \lambda\}$ to the TPA.

(6) ProofVerify Algorithm:

The TPA verifies the correctness of proof as follows:

$$e(\sigma, g) \overset{?}{=} e\left(g^x, g\right)^{\sum_{i \in i} v_i} \cdot e\left(\prod_{l=1}^{n} u_l^{ID}, g^{r_{ID}}\right)^{\sum_{i \in i} v_i} \cdot e\left(\prod_{i \in I} H(name \,||\, ID_i)^{v_i} \cdot \mu^{\lambda}, g^{rr}\right) \tag{1}$$

If the Eq. (1) holds, returns "1"; otherwise, returns "0". The process of integrity verification between the TPA and CSP is given in Fig. 2.



1. Generates a timestamp *t*;

2. Selects a random subset *I* with *l*-elements based on *t*, where $I \in [1,n]$.

3. Generates $v_i \in Z_P^*$ for each $i \in I$;

3. Determines $chal = \{i, v_i\}_{1 \le i \le n}$;

4. Computes $\lambda = \sum_{i \in I} b_{ij} v_i$;

5. Computes $\sigma = \prod_{i \in I} \sigma_i^{v_i}$;

6. Checks $e(\sigma,g) \overset{?}{=} e(g^x,g)^{\sum_{i \in i} v_i} \cdot e(\prod_{l=1}^{n} u_l^{ID}, g^{r_{ID}})^{\sum_{i \in i} v_i} \cdot e(\prod_{i \in I} H(name \,||\, ID_i)^{v_i} \cdot \mu^{\lambda}, g^{rr})$.

Figure 2 The process of integrity verification in our scheme

4 SECURITY ANALYSIS

4.1 Correctness and Auditing Soundness

In this section, the security of our proposed scheme is discussed in terms of correctness and audit soundness.

Theorem 1(Correctness). If the CSP stores the whole files of the user, the generated proof can pass the TPA's verification.

Proof: In *ProofVerify* algorithm, the TPA verifies the correctness of audit proof by checking what the Eq. (1) holds.

$$\begin{aligned}
e(\sigma, g) &= e\left(\prod_{i \in I} \sigma_i^{v_i}, g\right) = \\
&= e\left(\left(\prod_{i \in I} g^x \left(\prod_{l=1}^{n} u_l^{ID}\right)^{r_{ID}} \cdot \left(H(name \,||\, ID_i)\,\mu^{b_i}\right)^{rr}\right)^{v_i}, g\right) = \\
&= e\left(\left(\prod_{i \in I} \left(g^x\right)^{v_i}, g\right) \cdot e\left(\prod_{i \in I} \left(\prod_{l=1}^{n} u_l^{ID}\right)^{r_{ID} \cdot v_i}, g\right)\right) \cdot \\
&\quad \cdot e\left(\prod_{i \in I} \left(\left(H(name \,||\, ID_i)\,\mu^{b_i}\right)^{rr}\right)^{v_i}, g\right) = \\
&= e\left(\left(g^x\right)^{\sum_{i \in I} v_i}, g\right) \cdot e\left(\left(\prod_{l=1}^{n} u_l^{ID}\right)^{\sum_{i \in I} v_i}, g^{r_{ID}}\right) \cdot \\
&\quad \cdot e\left(\prod_{i \in I} H(name \,||\, ID_i)^{v_i} \cdot \prod_{i \in I} \left(\mu^{b_i}\right)^{v_i}, g^{rr}\right) = \\
&= e\left(g^x, g\right)^{\sum_{i \in I} v_i} \cdot e\left(\prod_{l=1}^{n} u_l^{ID}, g^{r_{ID}}\right)^{\sum_{i \in i} v_i} \cdot \\
&\quad \cdot\ e\left(\prod_{i \in I} H(name \,||\, ID_i)^{v_i} \cdot \left(\prod_{i \in I} \mu^{b_{ij}}\right)^{v_i}, g^{rr}\right) = \\
&= e\left(g^x, g\right)^{\sum_{i \in I} v_i} \cdot e\left(\prod_{l=1}^{n} u_l^{ID}, g^{r_{ID}}\right)^{\sum_{i \in i} v_i} \cdot \\
&\quad \cdot e\left(\prod_{i \in I} H(name \,||\, ID_i)^{v_i} \cdot \mu^{\sum_{i \in I} b_{ij} \cdot v_i}, g^{rr}\right) = \\
&= e\left(g^x, g\right)^{\sum_{i \in i} v_i} \cdot e\left(\prod_{l=1}^{n} u_l^{ID}, g^{r_{ID}}\right)^{\sum_{i \in i} v_i} \cdot \\
&\quad \cdot e\left(\prod_{i \in I} H(name \,||\, ID_i)^{v_i} \cdot \mu^{\lambda}, g^{rr}\right)
\end{aligned}$$

If the Eq. (1) holds, the TPA returns "1", which means the audit proof provided by the CSP is correct; otherwise, returns "0".

Theorem 2 (Audit soundness). For adversary $\mathcal{A}$ or an untrusted CSP, if the data blocks stored on the CSP have been corrupted, it is computationally infeasible to forge the valid audit proof *P* which can pass the TPA's verification.

Proof: In our scheme, the adversary $\mathcal{A}$ interacts with the challenger $\mathcal{C}$ many times to inquire and extract information. If the audit proof forged by the adversary $\mathcal{A}$ has passed the verifier's verification, we can extract all the challenged blocks by repeating the interaction between the proposed scheme and the constructed extractor. Through the following series of games, the audit soundness of our scheme can be proved.

Game 0. Game 0 is defined in section 3.

Game 1. Game 1 is the same as Game 0, with a minor difference that the challenger $\mathcal{C}$ keeps a list of all tags that adversary $\mathcal{A}$ has never queried. Whenever the adversary $\mathcal{A}$ makes *TagGen* query, the challenger $\mathcal{C}$ adds a record to the list.

Game 2. Game 2 is the same as Game 1, with one difference. The challenger $\mathcal{C}$ keeps a list of all responses from the adversary $\mathcal{A}$.

Given a valid audit proof $P = \{\lambda, \sigma\}$, it can successfully pass the verification of the following equation.

$$e(\sigma, g) = e\left(g^{x}, g\right)^{\sum_{i\in i} v_i} \cdot e\left(\prod_{l=1}^{n} u_l^{ID}, g^{rID}\right)^{\sum_{i\in i} v_i} \cdot \\ \cdot e\left(\prod_{i\in I} H\left(name \,||\, ID_i\right)^{v_i} \cdot \mu^{\lambda}, g^{rr}\right) \tag{2}$$

A proof $P' = \{\lambda', \sigma'\}$ forged by the adversary $\mathcal{A}$ as the output returns to the challenger $\mathcal{C}$. If the adversary $\mathcal{A}$ wins, but the tag $\sigma \neq \sigma'$, then this game aborts. It means that if the adversary $\mathcal{A}$ can successfully pass the verifier's checking, even if $\sigma \neq \sigma'$, the tag σ' still can pass the verification of the following equation, i.e.:

$$e(\sigma', g) = e\left(g^{x}, g\right)^{\sum_{i\in i} v_i} \cdot e\left(\prod_{l=1}^{n} u_l^{ID}, g^{rID}\right)^{\sum_{i\in i} v_i} \cdot \\ \cdot e\left(\prod_{i\in I} H\left(name \,||\, ID_i\right)^{v_i} \cdot \mu^{\lambda'}, g^{rr}\right) \tag{3}$$

Evidently, $\lambda \neq \lambda'$, otherwise $\sigma \neq \sigma'$, which contradicts our assumption. We define $\Delta\lambda = \lambda' - \lambda$, then we construct a simulator which can solve the CDH problem. Given $g, g^{\alpha}, h \in G_1$, the simulator's goal is output h^{α}. The simulator is equivalent to the challenger $\mathcal{C}$ in Game 1, but has the following differences:

(1) It randomly selects an element $x \in Z_p^*$, sets the master secret key $msk = x$, the master public key $mpk = g^{x}$. Then, it chooses two random values $a, b \in Z_p^*$ and sets $\mu = g^{a} h^{b}$.

(2) It generates the private key $sk_{ID} = \left(sk'_{ID}, sk''_{ID}\right) = \left(g^{rID}, g^{x} \cdot \left(\prod_{l=1}^{n} u_l^{ID}\right)^{rID}\right)$.

Dividing Eq. (3) by Eq. (2) and choosing a random value $\bar{x} \in Z_p^*$, then calculating the value $g^{rr} = (g^{\alpha})^{\bar{x}}$, we get

$$e(\sigma'/\sigma, g) = e((\mu^{\Delta\lambda}, g^{rr}) = e\left(\left(g^{a} h^{b}\right)^{\Delta\lambda}, \left(g^{\alpha}\right)^{\bar{x}}\right).$$

Further, we can get:

$$e\left(\sigma \cdot \sigma^{-1} \cdot \left(g^{b}\right)^{-\Delta\lambda\bar{x}a}, g\right) = e\left(h, g^{\alpha}\right)^{\Delta\lambda\bar{x}b} \tag{4}$$

Therefore, from the Eq. (4), we can know that $h^{\alpha} = (\sigma \cdot \sigma^{-1} \cdot (g^{b})^{-\Delta\lambda\bar{x}a})^{\frac{1}{\Delta\lambda\bar{x}b}}$. To analyze the probability that the challenger $\mathcal{C}$ aborts this game, we only need to calculate the probability that $\Delta\lambda\bar{x}b = 0 \, mod \, p$. The probability of $\Delta\lambda\bar{x}b = 0 \bmod p$ is $1/p$ which is negligible since p is a large prime. So, we can solve the CDH problem with the probability of $1 - 1/p$, which contradict assumption that the CDH problem in G_1 is computationally infeasible.

It shows that if the adversary wins the Game 1 and Game 2 with the non- negligible probability, the constructed simulator can solve the CDH problem.

Game 3. Game 3 is the same as Game 2, with one difference. The challenger $\mathcal{C}$ observes the proposed public auditing scheme. From the Game 2, we know that if the $\sigma \neq \sigma'$, the game will abort. In Game 3, if the $\lambda \neq \lambda'$, the challenger also will abort the game.

Given a valid integrity proof $P = \{\lambda, \sigma\}$, it can successfully pass the verification of the equation of $e(\sigma, g) = e\left(g^{x}, g\right)^{\sum_{i\in i} v_i} \cdot e\left(\prod_{l=1}^{n} u_l^{ID}, g^{rID}\right)^{\sum_{i\in i} v_i} \cdot e\left(\prod_{i\in I} H(name \,||\, ID_i)^{v_i} \cdot \mu^{\lambda}, g^{rr}\right) \cdot$ The proof $P' = \{\lambda', \sigma'\}$ forged by the adversary $\mathcal{A}$ as the output returns to the challenger $\mathcal{C}$. Because $\lambda \neq \lambda'$, this game aborts. It means that if the adversary $\mathcal{A}$ can successfully pass the verifier's verification, even if $\lambda \neq \lambda'$, the tag σ' still can pass the verification of the equation of

$$e(\sigma', g) = e\left(g^{x}, g\right)^{\sum_{i\in i} v_i} \cdot e\left(\prod_{l=1}^{n} u_l^{ID}, g^{rID}\right)^{\sum_{i\in i} v_i} \cdot \\ \cdot e\left(\prod_{i\in I} H\left(name \,||\, ID_i\right)^{v_i} \cdot \mu^{\lambda'}, g^{rr}\right).$$

We define $\Delta\lambda = \lambda' - \lambda$, and we have $1 = \mu^{\Delta\lambda}$. In this case, we have $\Delta\lambda \neq 0 \, mod \, p$. Otherwise, we have $\Delta\lambda = \lambda \, mod \, p$, which contradicts the aforementioned assumption.

Finally, from the above series of games, if the CSP can pass the TPA's verification, it must correctly store the user's complete data.

Theorem 3 (The Detectability). Assuming that encrypted file *F* is divided into *m* blocks, *d* denotes the bad blocks which may be corrupted or deleted by CSP, *c* is the challenged blocks in the audit process, and our scheme is $\left(\frac{d}{m}, 1 - \left(\frac{m-1}{m}\right)^{c}\right)$ detectable.

Proof: For *F*, we use a discrete random variable *X* to denote the number of the bad data blocks that is just match the challenged blocks. Let *PX* represent the probability that at least one bad block in the challenge set will be detected. So:

$$\begin{aligned} PX &= P\{X \geq 1\} = \\ &= 1 - P\{X = 0\} = \\ &= 1 - \frac{m-d}{m} \times \frac{m-1-d}{m-1} \times \ldots \times \frac{m-c+1-d}{m-c+1}. \end{aligned}$$

Because $1 - \left(\frac{m-1}{m}\right)^{c} \leq PX \leq \left(1 - \frac{m-c+1-d}{m-c+1}\right)^{c}$, we can get $PX \geq 1 - \left(\frac{m-1}{m}\right)^{c}$. So, the proposed scheme has the ability to detect the CSP's mis behavior $\left(\frac{d}{m}, 1 - \left(\frac{m-1}{m}\right)^{c}\right)$.

4.2 Against Attacks

Here, we discuss the security of proposed scheme against the three attacks.

Theorem 4. The proposed scheme can resist the replace attacks from the CSP.

Proof: Suppose that block b_i has been discarded or deleted by the CSP, but block b_k has remained, valid and uncorrupted. In the tag generation phase, the user *ID* computes tag σ_i for block b_i, $\sigma_i = g^x \cdot \left(\prod_{l=1}^{n} u_l^{ID}\right)^{r'ID} \cdot \left(H\left(name \| ID_i\right)\mu^{b_i}\right)^{rr}$.

And then it sends (b_i, σ_i) to the CSP. For some reason, the CSP replaces (b_i, σ_i) with (b_k, σ_k), tries to generate tag σ_k for block b_k, and expects σ_k to pass the TPA's correctness verification of block b_i.

Suppose that the blocks b_k are well remained on the cloud, where $k \in [1, n]$. After discarding or deleting the block b_i, the CSP tries to generate the tag σ_i' with the help of σ_k, i.e., $\sigma_i' = \alpha\sigma_k = \sigma_i$.

Since $\alpha\sigma_k = \alpha\left(g^x \cdot \left(\prod_{l=1}^{n} u_l^{ID}\right)^{r'ID} \cdot \left(H\left(name \| ID_k\right)\mu^{b_k}\right)^{rr}\right)$

And if $\sigma_i = \sigma_i'$.

It follows that $\sigma_i = \sigma_i' = \alpha\sigma_k$

$$g^x \cdot \left(\prod_{l=1}^{n} u_l^{ID}\right)^{r'ID} \cdot \left(H\left(name \| ID_i\right)\mu^{b_i}\right)^{rr} = \\ = \alpha \cdot g^x \cdot \left(\prod_{l=1}^{n} u_l^{ID}\right)^{r'ID} \cdot \left(H\left(name \| ID_k\right)\mu^{b_k}\right)^{rr} \tag{5}$$

If the Eq. (5) holds, the CSP must know $SK_{ID}'' = g^x \cdot \left(\prod_{l=1}^{n} u_l^{ID}\right)^{r'ID}$ and the following formula holds:

$$\left(H\left(name \| ID_i\right)\mu^{b_i}\right)^{rr} = \alpha\left(H\left(name \| ID_k\right)\mu^{b_k}\right)^{rr} \tag{6}$$

But in the proposed scheme, the tag is generated by the user *ID*, the CSP does not know how tags are constructed and the SK_{ID}'' is kept secret by the user *ID*. So, the probability that Eq. (5) holds is negligible, which means the CSP cannot pass the subsequent integrity audit with σ_i'.

So, if the CSP replaces (b_i, σ_i) with (b_k, σ_k), the replaced block cannot complete integrity verification instead of the original data block. Hence, the proposed scheme can resist replace attack.

Theorem 5. The proposed scheme can resist the forgery attacks from the CSP or TPA.

Proof: Suppose that there exists an adversary who modifies data block b_i to $b_i' = b_i + m_i$, $i \in [1, n]$. To pass the integrity verification, the adversary $\mathcal{A}$ must forge the tag for b_i' in the audit process.

Suppose that the tag σ' forged by adversary $\mathcal{A}$ for b' is as follows:

$$\sigma_i' = g^x \cdot \left(\prod_{l=1}^{n} u_l^{ID}\right)^{r'ID} \cdot \left(H\left(name \| ID_{i'}\right)\mu^{b_{i'}}\right)^{rr} \tag{7}$$

where $b_i' = b_i + m_i$. Given that $ID_{i'}$ is the identifier for block b', and $ID_{i'}$ is same with ID_i in principle.

Obviously, the probability that Eq. (7) is successfully calculated by any adversary $\mathcal{A}$ can be ignored. That is to say, the adversary $\mathcal{A}$ cannot pass the subsequent integrity audit with σ_i'. So, the proposed scheme can resist forgery attack from the CSP or TPA.

Theorem 6. The proposed scheme can resist the replay attacks from the CSP.

Proof: if the CSP has discarded or deleted b_i, it may attempt to execute the replay attack to pass the public audit by using another block b_i, but it is not possible. The proof process is similar to Theorem 4 and Theorem 5, so here it is omitted.

5 RESULTS AND CONCLUSION

5.1 Results

In this section, we evaluate the performance of our proposed scheme through several experiments. We run a series of experiments on a 1.8 GHz Intel Core i5 processor and 8GB RAM. All the experiments used the Type A with the free Pairing-Based Cryptography (PBC) Library. In the implementation, we choose the base filed size to be 512 bits, and the size of Z_p^* to be 160 bits, this is, $|p| = 160$ bits. The size of the data file is 20 MB.

To evaluate the performance of the proposed scheme, we set the number of the data blocks from 0 to 1000, increasing by an interval of 100. As shown in Fig. 3, the time cost of the extract and tag generation phase is given. According to Fig. 3, as the number of data blocks increases, the time it takes to generate the private key remains almost constant, and the curve shows a trend parallel to the *x* axis. But in the tag generation phase, the calculation time increases linearly with the increase in the number of data blocks. Fig. 4 shows the time overhead of the launch challenge and audit proof generation phases. We can see that initiating integrity challenge is extremely time saving, while generating audit proof is slightly time-consuming, and is directly related to the number of data blocks. The most important and time-consuming algorithm of the public auditing process is audit proof verification. In Fig. 5 and Fig. 6, we compare the overhead it takes to generate and verify audit proof with literature [19]. Although the computation time increases with the number of data blocks, our scheme is significantly less time-consuming than that in literature [19] and more effective.

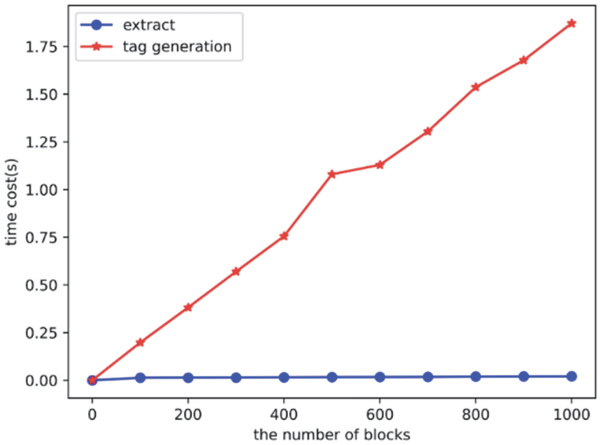


Figure 3 The time cost in the extract and tag generation phase

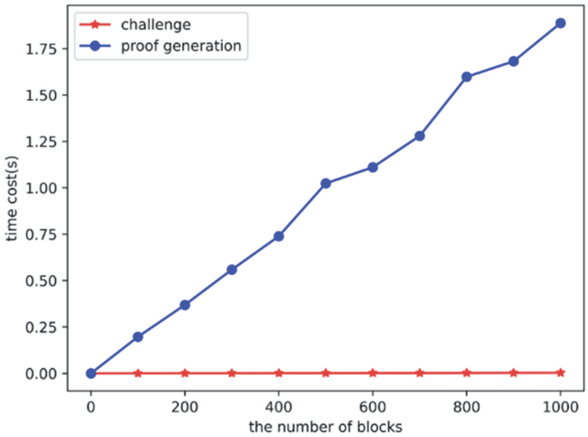


Figure 4 The time cost in challenge and proof generation phase

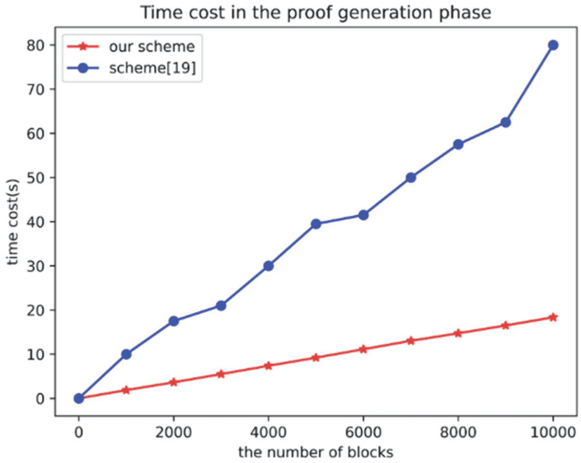


Figure 5 Comparison of the time cost in the proof generation phase between our scheme and scheme [19]

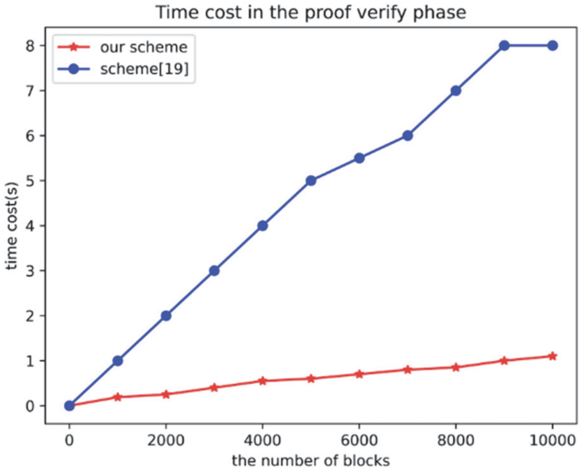


Figure 6 Comparison of the time cost in the proof verification phase between our scheme and scheme [19]

6 CONCLUSION

In this paper, we propose an identity-based public auditing scheme, which can not only solve the problem of user cloud data integrity checking but also can be used against malicious auditors. In our proposed scheme, to simplify complicated certificate management, the mathematical design is based on IBE. In addition, adding a timestamp in the public audit phase can prevent the TPA from dishonestly executing audit work. Besides, for untrusted CSP and semi-trusted TPA, the detailed security analysis proves that the proposed scheme is security. Finally, the effectiveness of the proposed scheme is validated. In future work, we will devote ourselves to exploring more methods to resist malicious auditors, so that the scheme can be better deployed in practice.

Acknowledgments

The financial supports from the National Natural Science Foundation of China (Grant No.: 61971014), the Key Project of Beijing Municipal Education Commission (Grant No.: SZ202110011006), and the Beijing Technology and Business University Young Teachers Research Startup Fund (Grant No.: QNJJ2021-54) are greatly acknowledged.

7 REFERENCES

[1] Husman, I. & Brezeanu, P. (2021). Progressive Taxation and Economic Development in EU Countries. A Panel Data Approach. *Economic Computation and Economic Cybernetics Studies and Research*, *55*(1), 285-300.

[2] Ateniese, G., Di Pietro, R., Mancini, L. V., &Tsudik, G. (2008). Scalable and efficient provable data possession. *In Proceedings of the 4th international conference on Security and privacy in communication netowrks*, 1-10. https://doi.org/10.1145/1460877.1460889

[3] Erway, C. C., Küpçü, A., Papamanthou, C., & Tamassia, R. (2015). Dynamic provable data possession. *ACM Transactions on Information and System Security (TISSEC)*, *17*(4), 1-29.https://doi.org/10.1145/2699909

[4] Ateniese, G., Burns, R., Curtmola, R., Herring, J., Khan, O., Kissner, L., & Song, D. (2011). Remote data checking using provable data possession. *ACM Transactions on Information and System Security (TISSEC)*, *14*(1), 1-34. https://doi.org/10.1145/1952982.1952994

[5] Wang, Q., Wang, C., Ren, K., Lou, W., & Li, J. (2010). Enabling public auditability and data dynamics for storage security in cloud computing. *IEEE transactions on parallel and distributed systems*, *22*(5), 847-859. https://doi.org/10.1109/TPDS.2010.183

[6] Curtmola, R., Khan, O., Burns, R., & Ateniese, G. (2008). MR-PDP: Multiple-replica provable data possession. *The 28th international conference on distributed computing systems*, 411-420.https://doi.org/ 10.1109/ICDCS.2008.68

[7] Barsoum, A. F. & Hasan, M. A. (2014). Provable multicopy dynamic data possession in cloud computing systems. *IEEE Transactions on Information Forensics and Security*, *10*(3), 485-497. https://doi.org/ 10.1109/TIFS.2014.2384391

[8] Wang, H. (2012). Proxy provable data possession in public clouds. *IEEE Transactions on Services Computing*, *6*(4), 551-559. https://doi.org/ 10.1109/TSC.2012.35

[9] Juels, A. & Kaliski Jr, B. S. (2007). PORs: Proofs of retrievability for large files. *Proceedings of the 14th ACM conference on Computer and communications security*, 584-597. https://doi.org/10.1145/1315245.1315317

[10] Shacham, H. & Waters, B. (2013). Compact proofs of retrievability. *Journal of cryptology*, *26*(3), 442-483. https://doi.org/10.1007/s00145-012-9129-2

[11] Wang, B., Li, B., & Li, H. (2014). Oruta: Privacy-preserving public auditing for shared data in the cloud. *IEEE transactions on cloud computing*, *2*(1), 43-56. https://doi.org/10.1109/TCC.2014.2299807

[12] Wang, B., Li, B., & Li, H. (2015). Panda: Public auditing for shared data with efficient user revocation in the cloud. *IEEE Transactions on services computing*, *8*(1), 92-106. https://doi.org/10.1109/TSC.2013.2295611

[13] Shen, J., Shen, J., Chen, X., Huang, X., & Susilo, W. (2017). An efficient public auditing protocol with novel dynamic structure for cloud data. *IEEE Transactions on Information Forensics and Security*, *12*(10), 2402-2415. https://doi.org/10.1109/TIFS.2017.2705620

[14] Kim, J. & Cho, Y. K. (2021). An assessment model for private contractor selection. Journal of Logistics. *Informatics and Service Science*, *8*(1), 35-50.

[15] Engel, M. M., Ramadhan, A., Abdurachman, E., & Trisetyarso, A. (2022). Mobile device security: a systematic literature review on research trends, methods and datasets. *Journal of System and Management Sciences*, *12*(2), 66-78.

[16] Zhang, Y., Xu, C., Yu, S., Li, H., & Zhang, X. (2015). SCLPV: Secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors. *IEEE Transactions on Computational Social Systems*, *2*(4), 159-170. https://doi.org/10.1109/TCSS.2016.2517205

[17] Wang, H., He, D., & Tang, S. (2016). Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud. *IEEE Transactions on Information Forensics and Security*, *11*(6), 1165-1176. https://doi.org/10.1109/TIFS.2016.2520886

[18] Zhang, J. & Dong, Q. (2016). Efficient ID-based public auditing for the outsourced data in cloud storage. *Information Sciences*, *343*, 1-14. https://doi.org/10.1016/j.ins.2015.12.043.

[19] Xue, J., Xu, C., Zhao, J., & Ma, J. (2019). Identity-based public auditing for cloud storage systems against malicious auditors via blockchain. *Science China Information Sciences*, *62*(3), 1-16. https://doi.org/10.1007/s11432-018-9462-0.

[20] Zhang, Y., Xu, C., Lin, X., & Xuemin, S. (2019). Blockchain-Based Public Integrity Verification for Cloud Storage Against Procrastinating Auditors. *IEEE Transactions on Cloud Computing*, *3*(9), 923-937. https://doi.org/10.1109/TCC.2019.2908400

[21] Zou, X., Deng, X., Wu, T. Y., & Chen, C. M. (2020). A collusion attack on identity-based public auditing scheme via Blockchain. *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, 97-105. https://doi.org/10.1007/978-981-13-9714-1_11

[22] Zhang, X., Zhao, J., Mu, L., Tang, Y., & Xu, C. (2019). Identity-based proxy-oriented outsourcing with public auditing in cloud-based medical cyber–physical systems. *Pervasive and Mobile Computing*, *56*, 18-28. https://doi.org/10.1016/j.pmcj.2019.03.004

[23] Yang, K. & Jia, X. (2013). An efficient and secure dynamic auditing protocol for data storage in cloud computing. *IEEE transactions on parallel and distributed systems*, *24*(9), 1717-1726. https://doi.org/10.1109/TPDS.2012.278

[24] Muntean, M., Brândaș, C., Cristescu, M., & Matiu, D. (2021). Improving Cloud Integration Using Design Science Research. *Economic Computation and Economic Cybernetics Studies and Research*, *55*(1), 201-218. https://doi.org/10.24818/18423264/55.1.21.13

[25] Gul, M. & Korkmaz, M. (2021). The Effect of Asymmetric and Symmetric Dependence in Some Growth Models Using Copulas. *Economic Computation and Economic Cybernetics Studies and Research*, *55*(1), 233-250. https://doi.org/10.24818/18423264/55.1.21.15

[26] Nam, J. & Choi, M. (2022). IoT edge cloud platform with Revocatable blockchain smart contract. *Journal of Logistics, Informatics and Service Science*, *9*(2), 131-144.

[27] Song, Y. J. & Lee, J. K. (2020). A blockchain-based fog-enabled energy cloud in internet of things. *Journal of Logistics, Informatics and Service Science*, *7*(2), 45-64.

[28] Seo, A., Son, Y., Lee, Y. S., & Jeong, J. H. (2022). A Blockchain Enabled Personal Donation System Development Scheme. *Journal of System and Management Sciences*, *12*(1), 103-119.

[29] Song, Y. J. & Lee, J. K. (2020). A Blockchain and Internet of Things Based Architecture Design for Energy Transaction. *Journal of System and Management Sciences*, *10*(2), 122-140. https://doi.org/10.33168/JSMS.2020.0209

Contact information:

Haojue ZHANG, PhD Lecturer
School of International Economics and Management,
Beijing Technology and Business University,
Beijing 100048, China
E-mail: haojuezhang@btbu.edu.cn

Yilin YUAN, PhD Lecturer
(Corresponding author)
School of Information Engineering,
Beijing Institute of Graphic Communication,
Beijing 102600, China
E-mail: yuanyilin@bigc.edu.cn

Xianwei XIN, PhD Lecturer
School of Computer and Information Engineering,
Henan Normal University,
Xinxiang, 453007, China
E-mail: xinxianwei@mail.bnu.edu.cn

Yanbo QU, PhD Candidate
Intelligent Manufacturing Electronics R&D Center,
Institute of Microelectronics of The Chinese Academy of Sciences,
Chaoyang District, Beijing, 100029, China
E-mail: quyanbo@ime.ac.cn