frontiers | Frontiers in Energy Research

TYPE Original Research
PUBLISHED 16 January 2023
DOI 10.3389/fenrg.2022.1058125

OPEN ACCESS

EDITED BY
Dou An,
MOE Key Laboratory for Intelligent Networks and Network Security, China

REVIEWED BY
Chengyu Hu,
Shandong University, China
Yalong Wu,
University of Houston—Clear Lake, United States

*CORRESPONDENCE
Wenting Shen,
shenwentingmath@163.com

SPECIALTY SECTION
This article was submitted to Smart Grids, a section of the journal Frontiers in Energy Research

RECEIVED 30 September 2022
ACCEPTED 31 October 2022
PUBLISHED 16 January 2023

CITATION
Gai C, Shen W, Yang M and Su Y (2023), Certificateless public auditing with data privacy preserving for cloud-based smart grid data.
Front. Energy Res. 10:1058125.
doi: 10.3389/fenrg.2022.1058125

Certificateless public auditing with data privacy preserving for cloud-based smart grid data

Chao Gai[1], Wenting Shen[1,2]*, Ming Yang[2] and Ye Su[3]

[1]College of Computer Science and Technology, Qingdao University, Qingdao, China, [2]Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China, [3]School of Information Science and Engineering, Shandong Normal University, Jinan, China

As the promising next generation power system, smart grid can collect and analyze the grid information in real time, which greatly improves the reliability and efficiency of the grid. However, as smart grid coverage expands, more and more data is being collected. To store and manage the massive amount of smart grid data, the data owners choose to upload the grid data to the cloud for storage and regularly check the integrity of their data. However, traditional public auditing schemes are mostly based on Public Key Infrastructure (PKI) or Identity Based Cryptography (IBC) system, which will lead to complicated certificate management and inherent key escrow problems. We propose a certificateless public auditing scheme for cloud-based smart grid data, which can avoid the above two problems. In order to prevent the disclosure of the private data collected by the smart grid during the phase of auditing, we use the random masking technology to protect data privacy. The security analysis and the performance evaluation show that the proposed scheme is secure and efficient.

KEYWORDS

smart grid, certificateless public auditing, cloud computing, cloud storage, privacy—preserving

1 Introduction

With the development of information technology, the smart grid becomes a new promising power system, which is allowed to collect and analyze smart grid data and provides more reliable, cost-effetive and efficient power management compared to traditional power grids (Chen et al., 2014; He et al., 2018a; Zhang et al., 2021c; Peng et al., 2021). A large amount of data are colleted with the expansion of smart grid coverage. Nevertheless, the traditional smart grid data management system without large storage space is unable to meet the data owners' storage requirements. Thus, more and more data owners choose to store smart grid data on the cloud.

Although the cloud provides a large amount of storage and computing resources for the data owners, there are some security issues that cannot be ignored (Zhang et al., 2021a; Yang et al., 2020; Lu et al., 2022; Shao et al., 2022). For example, the smart grid data stored in the cloud might be corrupted by hacker attacks, administrator's error operation, and damaged devices. Once the data is uploaded to the cloud, the data

owner will lose the physical control of the smart grid data stored in the cloud and cannot directly determine whether the data is intact or not. In order to ensure the integrity of cloud data, plenty of public auditing schemes are proposed (Ji et al., 2022; Li et al., 2021; Zhou et al., 2021; Liu et al., 2022). In public auditing, the data owner can delegate the data integrity auditing tasks to a Third Party Auditor (TPA) with abundant computation resources. In practice, the data collected by the smart grid might contain sensitive data, such as regional electricity consumption habits, residential electricity consumption patterns, etc (McDaniel and McLaughlin, 2009; Liu et al., 2021; Zhang et al., 2021b). Once the TPA is delegated to audit the data integrity, the data owner's data will inevitably be exposed to the TPA. The TPA is able to obtain sensitive information during the auditing phase. Therefore, it is critical to protect data privacy from the TPA in public auditing.

Nevertheless, most of the existing public auditing schemes are based on the traditional Public Key Infrastructure (PKI), which can lead to complex certificate management issue. In order to solve this problem, Identity Based Cryptography (IBC) had been proposed. In IBC system, there is a key generation center (KGC) which uses the data owner's identity to generate a private key for the data owner. The data owner can use his own identity as his public key. IBC eliminates the certificate management problem of PKI. However, the KGC holds the user's private key, the security of the user's private key will completely depend on the KGC, which leads to the inherent key escrow problem (Al-Riyami and Paterson, 2003; Wang et al., 2013). Therefore, in order to obtain better efficiency and higher security, the certificateless public key cryptography is proposed (Zhou et al., 2022; Zhang J et al., 2020; Xu et al., 2021). In certificateless public key cryptography systems, the data owner's private key is jointly generated by the KGC and the data owner. Therefore, the KGC does not know the data owner's complete private key. Certificateless public key cryptography can solve the inherent key escrow problem of IBC.

The contribution of our scheme can be summarized as follows:

1) Based on the certificateless public key cryptography, we proposed a certificateless public auditing scheme. Different from the existing public auditing schemes based on PKI or IBC, our scheme can avoid complex certificate management problem and key escrow problem.
2) To achieve data privacy preserving, we utilize the novel random masking technology in the phase of auditing. The TPA cannot obtain the sensitive data from the proof generated by the cloud.
3) We give the security proof of the proposed scheme. Furthermore, the theoretical analysis and experimental results show that the proposed scheme is efficient.

1.1 Related work

Ateniese et al. (2007) proposed the first "Provable Data Possession" (PDP) scheme, in which the integrity of the remote data can be checked by the client. In this scheme, the homomorphic authenticators and the random sampling technique are employed to achieve the data integrity checking. Juels and Kaliski (2007) constructed a "Proofs of Retrievability" (PoR) scheme, which guarantees the data integrity and data retrievability on the cloud. However, in this scheme, the verifier can only perform a finite number of data integrity verification. In 2008, Shacham and Waters (2008) designed an improved PoR scheme, which is provably secure.

To support data dynamic, Ateniese et al. (2008) constructed a PDP scheme supporting data dynamic operations. Guo et al. (2020) designed a dynamic proof of data possession and replication scheme. In this scheme, the multiple replicas share a single authenticated tree. Erway et al. (2015) designed a rank-based skip list and constructed the PDP scheme supporting full data dynamic operations. Wang et al. (2019) proposed a blockchain-based private data integrity verification scheme by using RSA signature. Wang et al. (2017b) designed a cloud storage auditing scheme based on the online/offline signature, in which the data owner can reduce the burden of authenticator generation in the online phase. To improve the auditing efficiency, Gao et al. (2021) designed a data integrity checking scheme based on the keyword. This scheme allows the TPA to verify the integrity of files containing the specified keyword. To preserve the data privacy, Li et al. (2018) designed a privacy-preserving data integrity verification scheme with zero-knowledge proof. In addition, there are many researches devoted to the key exposure problem (Yu et al., 2016; Yu and Wang, 2017; Xu et al., 2020).

To solve complex certificate management, in 1984, Identity Based Cryptography (IBC) is proposed by Shamir (1985). In IBC system, the data owner's private key is calculated by a trusted Key Generation Center (KGC) with the data owner's identity. The data owner uses his identity as the public key, which eliminates the complex certificate management. Wang et al. (2014) proposed the first identity-based data integrity checking scheme by using the Schnorr signature. To support efficient user revocation, Zhang Y et al. (2020) designed an identity-based data integrity verification scheme for shared data. Shen et al. (2019) proposed a data integrity checking scheme supporting data sharing and sensitive data hiding. In their scheme, the data owner's sensitive data can be protected under the assistance of the sanitizer. Wang et al. (2017a) designed an identity-based comprehensive data integrity checking scheme, in which the authenticators can be generated with the help of the proxy. To protect the data owner's identity privacy, Zhang et al. (2019) utilized the anonymous identity to replace the data owner's real identity, and constructed a

conditinal identity privacy-preserving data integrity checking scheme.

Unfortunately, although the IBC system can avoid the certificate management problem caused by PKI, it still has the inherent key escrow issue. Zhang et al. (2015) designed a secure certificateless public data integrity verification scheme, which can resist the malicious TPA. He et al. (2018b) proposed a certificateless data integrity auditing scheme which can resist the attacks of two types of adversaries in certificateless crytography (The adversary is able to replace the public keys of the users and the adversary is able to access the master key of the KGC). To eliminate the problem of key escrow in IBC, Wu et al. (2019) proposed a certificateless public auditing scheme which supports identity privacy protection. Zhou et al. (2022) applied the certificateless technology to the multi-replica environment. This scheme can realize the efficient data dynamic in the multi-replica environment by using the new Merkle Hash Tree structure.

1.2 Organization

The remainder of this paper is organized as follows: In **Section 2**, we introduce the system model and design goals of our scheme; In **Section 3**, we describe the preliminaries and definition; We give a detailed algorithm of our scheme in **Section 4**; In **Section 5**, we analyzed the security of our scheme; We show the performance analysis of our scheme in **Section 6**; We make a conclusion in **Section 7**.

2 System model and design goals

We give the system model and the design goals in this section.

2.1 System model

As shown in **Figure 1**, the system model of our scheme contains four entities: the cloud, the data owner, the Key Generation Center (KGC) and the Third Party Auditor (TPA).

1) The cloud: The cloud is an entity which has enormous storage and computation resources. It is responsible for storing and managing the smart grid data for the data owner. After receiving the TPA's challenge, the cloud needs to send the corresponding auditing proof to the TPA.
2) The data owner: The data owner is an entity with limited storage space and computation resources. He outsources his smart grid data to the cloud for storage and delegates the TPA to verify the integrity of cloud data.

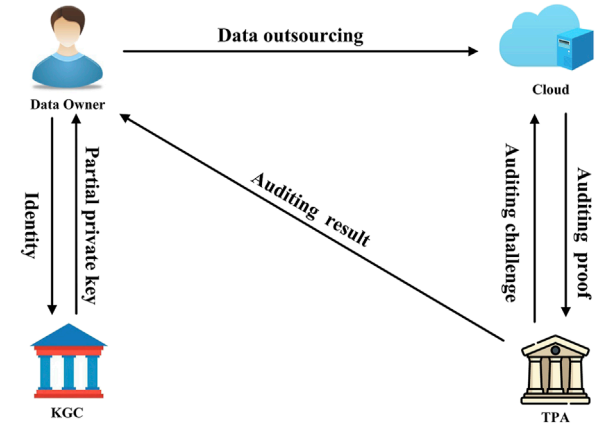

FIGURE 1
System model.

3) KGC: The KGC is an entity which takes charge of producing the system parameters and calculating the partial private key for the data owner based on the data owner's identity.
4) TPA: The TPA is an entity with powerful computing capabilities, which needs to generate and deliver the auditing challenge to the cloud and audit the integrity of the cloud data.

2.2 Design goals

In order to achieve privacy preserving in certificateless public auditing for cloud-based smart grid data, our scheme needs to meet the following design goals:

1) Correctness: If the KGC generates the partial private key for the data owner honestly, the partial private key can pass the data owner's checking. If the cloud generates the auditing proof honestly, the auditing proof can pass the TPA's checking.
2) Soundness: The cloud cannot pass the TPA's verification if the data has been corrupted.
3) Privacy protection: In the phase of data integrity auditing, the TPA cannot obtain the smart grid data from the cloud's auditing proof.

3 Preliminaries and definition

In this section, we present the preliminaries applied in our scheme. Then, we give the definitions of our scheme.

3.1 Preliminaries

3.1.1 Bilinear pairing

Suppose that there are two different multiplicative cyclic groups G_1 and G_T with the same prime order p. The generator of the group G_1 is g. If e:$G_1 \times G_1 \rightarrow G_T$ is a bilinear pairing, it satisfies (Boneh et al., 2001):

i) Bilinearity: For $\forall u, v \in G_1$ and $\forall \alpha, \beta \in Z_p^*$, we have $e(u^\alpha, v^\beta) = e(u,v)^{\alpha\beta}$.
ii) Non-degeneracy: $\exists u, v \in G_1$ and $e(u,v) \neq 1_{G_T}$.
iii) Computability: For $\forall u, v \in G_1$, $e(u,v)$ is able to be computed efficiently.

3.1.2 Computational diffie-hellman problem

Given $g, g^\alpha, g^\beta \in G_1$, where g is the generator of G_1 and $\alpha, \beta \in Z_p^*$, compute $g^{\alpha\beta} \in G_1$. The CDH assumption in G_1 holds if it is hard to solve the CDH problem in G_1 (Bao et al., 2003).

3.1.3 Discrete logarithm problem

Given $g, g^\alpha \in G_1$, where g is the generator of G_1 and $\alpha \in Z_p^*$, compute α. The DL assumption in G_1 holds if it is hard to solve the DL problem in G_1 (McCurley, 1990).

3.2 Definition

Definition 1: Our scheme includes seven algorithms: Setup, PartialKeyGen, PrivateKeyGen, AuthenticatorGen, ChallengeGen, ProofGen and ProofVerify.

1) Setup $(1^\lambda) \rightarrow (sk_K, params)$: This algorithm is run by the KGC. Taking λ as input, it outputs the KGC's master secret key sk_K and the system parameters *params*.
2) PartialKeyGen $(ID_O, sk_K, params) \rightarrow (\sigma_O)$: This algorithm is run by the KGC and the data owner. Inputting the data owner's identify ID_O, the KGC's master secret key sk_K and the system parameters *params*, it outputs the data owner's partial private key σ_O. The data owner can check whether the partial private key σ_O is valid or not.
3) PrivateKeyGen $(\sigma_O, params) \rightarrow (sk_O, pk_O)$: This algorithm is run by the data owner. Taking the partial private key σ_O and the system parameters *params* as input, it outputs the data owner's private key sk_O and the corresponding public key pk_O.
4) AuthenticatorGen $(sk_O, F, ID_F) \rightarrow (T, tag)$: This algorithm is run by the data owner. Inputting the data owner's private key sk_O, the file F and the file's identifier ID_F, it generates the authenticator set T and the file tag *tag*.
5) ChallengeGen $(s, K_1, K_2) \rightarrow (chal)$: This algorithm is run by the TPA. Taking three random values s, K_1 and K_2 as input, it produces the auditing challenge *chal*.
6) ProofGen $(chal, F, T) \rightarrow (proof)$: This algorithm is run by the cloud. Taking the challenge *chal*, the file F and the authenticator set T as input, it outputs the auditing proof *proof*.
7) ProofVerify $(tag, proof, pk_K, pk_O) \rightarrow 0,1$: This algorithm is run by the TPA. Inputting the file tag *tag*, the auditing proof *proof*, the KGC's public key pk_K and the data owner's public key pk_O, it outputs the auditing result. If the proof is valid, the result is "1"; otherwise, the result is "0."

4 The proposed scheme

In this section, we give the detailed algorithms of our scheme.

1) Setup $(1^\lambda) \rightarrow (sk_K, params)$
   a) Let e be a bilinear pairing e:$G_1 \times G_1 \rightarrow G_T$, where G_1 and G_T are two different cyclic multiplicative groups with the same prime order p. The KGC selects two independent generators g and u of the group G_1 and sets two different hash functions: H:$\{0.1\}^* \rightarrow G_1$ and H_1:$\{0.1\}^* \rightarrow G_1$.
   b) The KGC randomly picks an element $sk_K \in Z_p^*$ as its master secret key, generates $pk_K = g^{sk_K}$ as its system public key, and publishes the system parameter $params = (G_1, G_T, e, g, u, pk_K, H, H_1)$.
2) PartialKeyGen $(ID_O, sk_K, params) \rightarrow (\sigma_O)$
   a) The data owner transmits his identify $ID_O \in \{0,1\}^*$ to the KGC.
   b) The KGC computes $\sigma_O = H(ID_O)^{sk_K}$, and transmits σ_O to the data owner as his partial private key.
   c) After receiving σ_O, the data owner checks whether the partial private key is correct or not according to the following equation

$$e(\sigma_O, g) = e(H(ID_O), pk_K) \tag{1}$$

If **Equation 1** holds, the data owner accepts the partial private key σ_O.

3) PrivateKeyGen $(\sigma_O, params) \rightarrow (sk_O, pk_O)$

The data owner picks a random value $x \in Z_p^*$ and sets $sk_O = \{x, \sigma_O\}$ as his private key. The data owner calculates $pk_O = g^x$ as his public key.

4) AuthenticatorGen $(sk_O, F, ID_F) \rightarrow (T, tag)$
   a) The data owner divides the file F into n data blocks d_i $(i \in [1,n])$. The data owner generates the corresponding authenticator $t_i = \sigma_O \cdot (H_1(ID_F \| i \| n) u^{d_i})^x$ for each data block d_i $(i \in [1,n])$, where ID_F is the identifier of the file F. The file F's authenticator set is denoted as $T = \{t_i\}_{1 \le i \le n}$.
   b) The data owner produces a file tag $tag = ID_F \| n \| SSig_{ssk}(ID_F \| n)$ using the signature *SSig*, where *ssk* is the private key of the signature *SSig*.
   c) The data owner uploads the file F, the authenticator set T and the file tag *tag* to the cloud.

5) ChallengeGen $(s, K_1, K_2) \rightarrow (chal)$
   a) For each challenge, the TPA randomly picks three values s ($s \in [1, n]$) and $K_1, K_2 \in Z_p^*$, where K_1 is the key of pseudo-random permutation $\pi_{K_1}(\cdot)$ and K_2 is the key of pseudo-random function $\phi_{K_2}(\cdot)$.
   b) The TPA sends the challenge $chal = \{s, K_1, K_2\}$ to the cloud.
6) ProofGen $(chal, F, T) \rightarrow (proof)$
   a) According to the challenge *chal*, the cloud generates the challenged block's index $l_j = \pi_{K_1}(j)$ for each $1 \le j \le s$, where $l_j \in [1, n]$.
   b) The cloud calculates a random value $v_j = \phi_{K_2}(j)$ for each $1 \le j \le s$, in which $v_j \in Z_p^*$.
   c) The cloud computes $\Gamma = \prod_{j=1}^{s} t_{l_j}{}^{v_j}$, $\mu' = \sum_{j=1}^{s} d_{l_j} v_j$.
   d) In order to protect the data privacy, the cloud chooses a random element $r \in Z_p^*$ and computes $\mu = \mu' - r$ to blind μ'. The cloud calculates $R = u^r$.
   e) The cloud transmits the proof $proof = (\mu, \Gamma, R)$ and the file tag *tag* to the TPA.
7) ProofVerify $(tag, proof, pk_K, pk_O) \rightarrow 0.1$
   a) The TPA verifies the validity of the file tag *tag*. If *tag* is valid, the TPA parses the file's identifier ID_F and the number of data blocks n.
   b) For each $1 \le j \le s$, the TPA calculates $l_j = \pi_{K_1}(j)$ and $v_j = \phi_{K_2}(j)$.
   c) The TPA verifies whether the auditing proof*proof* is valid or not according to the following equation

$$e(\Gamma, g) = e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu} \cdot R, pk_O\right) \quad (2)$$

If the above equation holds, the TPA returns "1", which means that the remote data is intact; otherwise, it returns "0", which means that the remote data is broken.

5 Security analysis

In this section, we give the security proof of our scheme from the aspects of correctness, soundness and data privacy protection.

5.1 Theorem 1 (Correctness)

In our scheme, if the KGC, the TPA, and the cloud honestly perform the specified procedures, the partial private key and the auditing proof are able to pass the verification.

5.1.1 Proof

The derivation process for the data owner to verify whether the partial key is correct is as follows:

$$\begin{aligned} e(\sigma_O, g) &= e\left(H(ID_O)^{sk_K}, g\right) \\ &= e\left(H(ID_O), g^{sk_K}\right) \\ &= e\left(H(ID_O), pk_K\right) \end{aligned}$$

The derivation process for the TPA to verify whether the auditing proof is valid is as follows:

$$\begin{aligned} & e(\Gamma, g) \\ &= e\left(\prod_{j=1}^{s} t_{l_j}{}^{v_j}, g\right) \\ &= e\left(\prod_{j=1}^{s} \left(\sigma_O \cdot \left(H_1\left(ID_F \| l_j \| n\right) u^{d_{l_j}}\right)^{x}\right)^{v_j}, g\right) \\ &= e\left(\prod_{j=1}^{s} \sigma_O{}^{v_j} \cdot \prod_{j=1}^{s} \left(H_1\left(ID_F \| l_j \| n\right) u^{d_{l_j}}\right)^{x v_j}, g\right) \\ &= e\left(\prod_{j=1}^{s} \sigma_O{}^{v_j}, g\right) \cdot e\left(\prod_{j=1}^{s} \left(H_1\left(ID_F \| l_j \| n\right) u^{d_{l_j}}\right)^{x v_j}, g\right) \\ &= e\left(\sigma_O{}^{\sum_{j=1}^{s} v_j}, g\right) \cdot e\left(\prod_{j=1}^{s} \left(H_1\left(ID_F \| l_j \| n\right) u^{d_{l_j}}\right)^{v_j}, g^{x}\right) \\ &= e\left(H(ID_O)^{sk_K \cdot \sum_{j=1}^{s} v_j}, g\right) \\ &\quad \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot \prod_{j=1}^{s} u^{d_{l_j} v_j}, pk_O\right) \\ &= e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, g^{sk_K}\right) \\ &\quad \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\sum_{j=1}^{s} d_{l_j} v_j}, pk_O\right) \\ &= e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \\ &\quad \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu'}, pk_O\right) \\ &= e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \\ &\quad \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu + r}, pk_O\right) \\ &= e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \\ &\quad \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu} \cdot u^{r}, pk_O\right) \end{aligned}$$

$$= e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu} \cdot R, pk_O\right)$$

5.2 Theorem 2 (Soundness)

Suppose the CDH assumption holds in G_1 and the signature scheme used for generating tag is existentially unforgeable. In our scheme, for an adversary, it is computationally infeasible to generate a bogus proof that is able to pass the TPA's checking if the cloud data has been damaged.

Proof. We will prove this theorem with the method of knowledge proof. The malicious cloud is viewed as adversary and the user plays the role on the challenger.

Game 0. If the adversary submits one tag, the challenger will abort if the tag is a valid *SSig* signature but not signed by the challenger.

Analysis. If the challenger aborts in Game 0 with non-negligible probability, the adversary is able to forge a valid *SSig* signature. This contradicts the assumption that *SSig* is an unforgeable signature. Therefore, the file identifier and the number of data blocks in the interactions with the adversary are all valid and generated by the challenger.

Game 1. Game 1 is the same as Game 0, with only one difference. The challenger keeps a list of his responses to the queries from the adversary. If the adversary wins the game 1 but the aggregated authenticator Γ^* in the proof is different from $\Gamma = \prod_{j=1}^{s} t_{l_j}^{v_j}$, then the challenger will abort.

Analysis. Assume $proof = (\mu, \Gamma, R)$ is a valid proof. We have:

$$e(\Gamma, g) = e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu} \cdot R, pk_O\right) \quad (3)$$

Suppose $proof^* = (\mu^*, \Gamma^*, R)$ is a forged auditing proof, where Γ^* is different from Γ. Because the forgery is successful, $proof^*$ can pass the verification of the following equation:

$$e\left(\Gamma^*, g\right) = e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu^*} \cdot R, pk_O\right) \quad (4)$$

It is obviously that $\mu \neq \mu^*$; otherwise $\Gamma = \Gamma^*$, which contradicts the above assumption. Let $\Delta\mu = \mu^* - \mu$. The adversary can win the game 1 with a non-negligible probability only if there is a simulator can solve the CDH problem.

Given $g, g^{\varepsilon}, h \in G_1$, the simulator needs to generate h^{ε}. The simulator picks two random values $\beta, \theta \in Z_p^*$ and sets $u = g^{\beta} h^{\theta}$ and $pk_O = g^{\varepsilon}$. For each l_j, the simulator selects a random value v_j and programs a random oracle at l_j as $H_1(ID_F \| l_j \| n) = g^{v_j} / (g^{\beta} h^{\theta})^{d_{l_j}}$. So, we can obtain $H_1(ID_F \| l_j \| n) u^{d_{l_j}} = g^{v_j} / (g^{\beta} h^{\theta})^{d_{l_j}} \cdot (g^{\beta} h^{\theta})^{d_{l_j}} = g^{v_j}$.

Dividing **Eq. 4** by **Eq. 3**, we have

$$\begin{aligned} & e\left(\Gamma^* / \Gamma, g\right) \\ &= e\left(u^{\mu^* - \mu}, pk_O\right) \\ &= e\left(\left(g^{\beta} h^{\theta}\right)^{\Delta\mu}, pk_O\right) \\ &= e\left(g^{\beta\Delta\mu}, pk_O\right) \cdot e\left(h^{\theta\Delta\mu}, pk_O\right) \\ &= e\left(g^{\beta\Delta\mu}, g^{\varepsilon}\right) \cdot e\left(h^{\theta\Delta\mu}, g^{\varepsilon}\right) \\ &= e(g,g)^{\beta\Delta\mu\varepsilon} \cdot e(h^{\varepsilon}, g)^{\theta\Delta\mu} \end{aligned}$$

According to the above equation, we can obtain $e(\Gamma^* / \Gamma \cdot g^{-\beta\Delta\mu\varepsilon}, g) = e(h^{\varepsilon}, g)^{\theta\Delta\mu}$. So, we have $h^{\varepsilon} = (\Gamma^* / \Gamma \cdot g^{-\beta\Delta\mu\varepsilon})^{\frac{1}{\theta\Delta\mu}}$.

The probability of $\theta\Delta\mu \neq 0$ is $1 - \frac{1}{p}$, which is non-negligible. So, we can solve the CDH problem with the probability $1 - \frac{1}{p}$, which is contradiction with the assumption that the CDH problem in G_1 is computationally infeasible.

Game 2. Game 2 is similar to Game 1, with one difference. The challenger records all interactions with the adversary.

If the adversary wins the game 2 but the aggregated data block μ^* in the proof is different from the expected one μ, then the challenger will abort.

Analysis. Suppose $proof = (\mu, \Gamma, R)$ is a valid proof. We get:

$$e(\Gamma, g) = e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu} \cdot R, pk_O\right)$$

Assume $proof^* = (\mu^*, \Gamma^*, R)$ is a forged auditing proof. Because the forgery is successful, we get:

$$e\left(\Gamma^*, g\right) = e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu^*} \cdot R, pk_O\right)$$

Based on Game 1, we have $\Gamma = \Gamma^*$. Set $\Delta\mu = \mu^* - \mu$, we can design a simulator to solve the DL problem.

Inputting $g, h \in G_1$, the simulator needs to output ε satisfying $h = g^{\varepsilon}$. The simulator selects two random values $\beta, \theta \in Z_p^*$ and

sets $u = g^{\beta}h^{\theta}$. Based on $\Gamma = \Gamma^{*}$, we obtain

$$\begin{aligned} & e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \\ & \quad \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu} \cdot R, pk_O\right) \\ & = e(\Gamma, g) \\ & = e\left(\Gamma^{*}, g\right) \\ & = e\left(H(ID_O)^{\sum_{j=1}^{s} v_j}, pk_K\right) \\ & \quad \cdot e\left(\prod_{j=1}^{s} H_1\left(ID_F \| l_j \| n\right)^{v_j} \cdot u^{\mu^{*}} \cdot R, pk_O\right) \end{aligned}$$

Further, we obtain that $u^{\mu} = u^{\mu^{*}}$ and $1 = u^{\Delta\mu} = (g^{\beta}h^{\theta})^{\Delta\mu} = g^{\beta\Delta\mu}h^{\theta\Delta\mu}$. Hence, we can solve the DL problem as follow:

$$h = g^{-\frac{\beta\Delta\mu}{\theta\Delta\mu}} = g^{-\frac{\beta}{\theta}}$$

The probability of $\theta \neq 0$ is $1 - \frac{1}{p}$, and it is non-negligible. So, we can solve the DL problem with the non-negligible probability $1 - \frac{1}{p}$, which is contradiction with the assumption that the DL problem in G_1 is computationally infeasible. Therefore, if the cloud can pass the TPA's verification with non-negligible probability, it means that the cloud correctly stores the smart grid data.

5.3 Theorem 3 (data privacy protection)

In our scheme, the TPA cannot extract the real data from the cloud's auditing proof.

Proof. On the one hand, in the auditing proof $= (\mu, \Gamma, R)$, the original aggregated data block $\mu' = \sum_{j=1}^{s} d_{l_j} v_j$ is blinded as μ by the random value r, where $\mu = \mu' - r$. Because the DL problem in G_1 is hard, the TPA cannot extract the value r from R, where $R = u^{r}$. Thus, the TPA cannot obtain the original aggregated data block μ' from μ. On the other hand, we get that

$$\begin{aligned} \Gamma &= \prod_{j=1}^{s} t_{l_j}{}^{v_j} = \prod_{j=1}^{s} \left(\sigma_O \cdot \left(H_1\left(ID_F \| l_j \| n\right) u^{d_{l_j}}\right)^{x}\right)^{v_j} \\ &= \prod_{j=1}^{s} \left(\sigma_O \cdot H_1\left(ID_F \| l_j \| n\right)^{x}\right)^{v_j} \cdot \left(u^{\mu'}\right)^{x} \end{aligned}$$

From the above equation, we know that $(u^{\mu'})^{x}$ is blinded by $\prod_{j=1}^{s} (\sigma_O \cdot H_1(ID_F \| l_j \| n)^{x})^{v_j}$. It is computational infeasible to compute $\prod_{j=1}^{s} (\sigma_O \cdot H_1(ID_F \| l_j \| n)^{x})^{v_j}$ from $\prod_{j=1}^{s} (\sigma_O \cdot H_1(ID_F \| l_j \| n))^{v_j}$ and g^{x} because the CDH problem in G_1 is hard. So, the TPA cannot get $(u^{\mu'})^{x}$ from Γ. Consequently, the TPA cannot obtain the real smart grid data during the auditing phase.

TABLE 1 Computation overhead in different phases.

Phase	Computation overhead
Partial key generation	$(2P + 2H + E)$
Authenticator generation	$n(H + 2M + 2E)$
Proof generation	$(2s - 1)M + sE + (s - 1)A$
Proof verification	$3P + (s + 1)(H + M) + (s + 2)E + (s - 1)A$

TABLE 2 Communication overhead.

Phase	Communication overhead
Partial key generation	$\|q\| + \|p\|$
Data outsourcing	$n\|q\| + (n + 1)\|p\|$
Data integrity auditing	$\|n\| + 2\|q\| + 3\|p\|$

6 Performance analysis

In this section, we systematically analyze the performance of our scheme from both theoretical analysis and experimental results.

6.1 Theoretical analysis

We respectively use P, H, M, E, and A to denote one pairing operation, one hash operation, one multiplication operation, one exponentiation operation and one addition operation. Suppose that the file is divide into n data blocks, and the TPA challenges s data blocks. In Table 1, we describe the computation overhead of our scheme in different phases. In the phase of partial key generation, the computation overhead is $(2P + 2H + E)$. In the phase of authenticator generation, the user requires n $(H + 2M + 2E)$ computation overhead to generate the authenticators. In the phase of proof generation, the computation overhead on the cloud side is $(2s - 1)M + sE + (s - 1)A$. In the phase of proof verification, the TPA needs to cost $3P + (s + 1)(H + M) + (s + 2)E + (s - 1)$ to verify the auditing proof.

Let $|q|$, $|p|$ and $|n|$ be the size of an element in G_1, Z_p^{*} and set $[1, n]$ respectively. We present the communication overhead of our scheme in Table 2. The communication overhead of partial key generation is $|q| + |p|$. The communication overhead of data outsourcing is $n|q| + (n + 1)|p|$. In the data integrity auditing phase, the communication overhead is $|n| + 2|q| + 3|p|$.

6.2 Experimental results

In order to show the performance of our scheme, we design a series of experiments to simulate our scheme. We utilize C programming language with the GNU Multiple

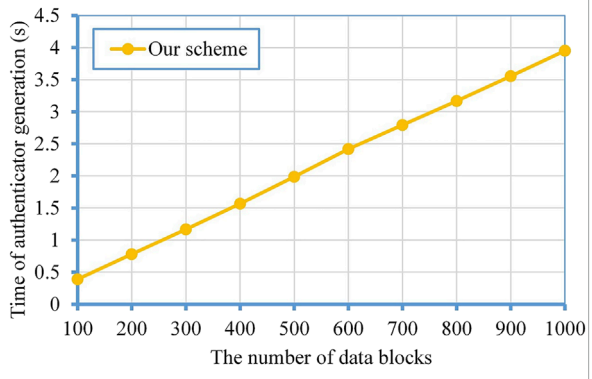

FIGURE 2
The computation overhead of authenticator generation.

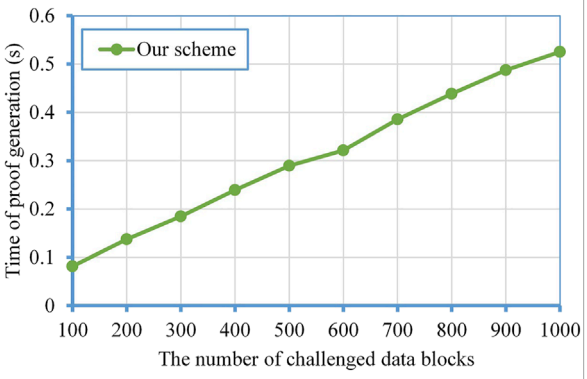

FIGURE 4
The computation overhead of proof generation.

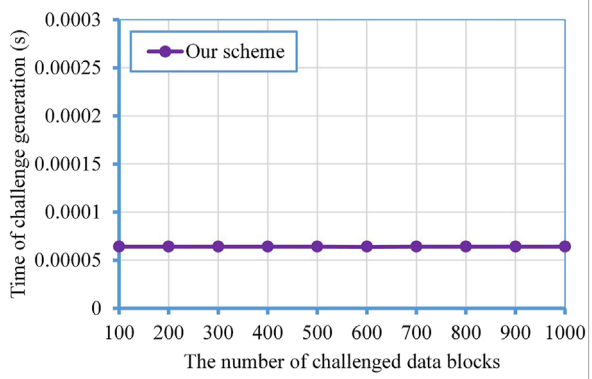

FIGURE 3
The computation overhead of challenge generation.

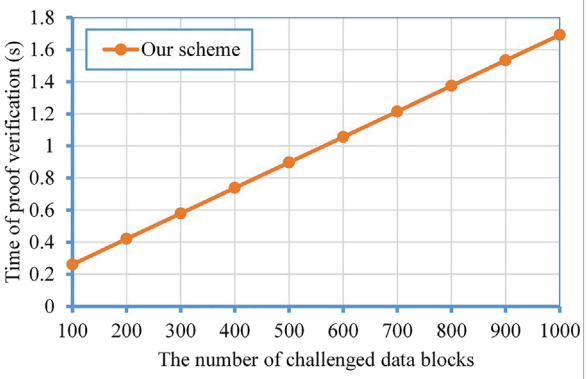

FIGURE 5
The computation overhead of proof verification.

Precision Arithmetic (GMP) Library (GMP-6.2.1) (GMP, 1991) and the Pairing Based Cryptography (PBC) Library (PBC-0.5.14) (Lynn, 2015) to implement the experiments. The experiments are conducted on the Ubuntu 20.04 (4 GB memory) VMware 16.1 pro in a desktop running Windows 10 with Intel(R) Core (TM) i7-9700T @ 2.0 GHZ and 16 GB RAM. In the following experiments, we set the base field size to be 512 bits, the size of an element in Z_p^* to be $|p| = 160$ bits.

6.2.1 Authenticator generation

In the proposed scheme, we test the time of authenticator generation for different numbers of data blocks, ranging from 100 to 1,000. The experimental result is represented in **Figure 2**. When the data owner outsources 100 data blocks and calculates the authenticators for these data blocks, the time to generate the authenticators is 0.390975s. When the number of data blocks is 1,000, the time of authenticator generation is 3.951747s. We can find that the time of authenticator generation is related to the number of data blocks.

6.2.2 Challenge generation

In the following experiment, we set the number of data blocks to 1,000, and the number of queried data blocks ranges from 100 to 1,000. As shown in **Figure 3**. When the number of challenged data blocks is 100, the time of challenge generation is 0.000064s. And if the number of challenged data blocks is 1,000, the challenge generation time is 0.000063s. Obviously, the overhead of challenge generation is independent of the number of challenged data blocks.

6.2.3 Proof generation

In **Figure 4**, we can obtain that the computation overhead of proof generation increases linearly with the number of challenged data blocks. When the number of queried data blocks is 100, the proof generation time is 0.081322s. When

the number of challenged data blocks is 1,000, the time cost is 0.525229s.

6.2.4 Proof verification

Figure 5 shows that there is a proportional relationship between the computation cost of proof verification and the number of challenged data blocks. As the number of challenged data blocks increases from 100 to 100, the time of proof verification increases from 0.261052s to 1.691584s.

7 Conclusion

In this paper, we proposed a certificateless public auditing scheme for cloud-based smart grid data, which supports data privacy preserving. Compare with the traditional public auditing schemes based on PKI or IBC, our scheme can avoid the complex certificate management issue and key escrow isssue. In addition, the TPA cannot obtain the original smart grid data during the data integrity auditing phase. We give the security proof of the scheme, and the results show that our scheme is secure. We also evaluate the efficiency of our scheme through a series of experiments.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

CG organized the manuscript and wrote the first draft of the manuscript, WS provided revisions to this paper, MY and YS designed experimental scenarios.

Funding

This research is supported by National Natural Science Foundation of China (62102211), Shandong Provincial Natural Science Foundation, China (ZR2021QF018), the Open Research Fund from Shandong Key Laboratory of Computer Network (SDKLCN-2020-02).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Al-Riyami, S. S., and Paterson, K. G. (2003). "Certificateless public key cryptography," in *Advances in cryptology - asiacrypt 2003*. Editor C.-S. Laih (Berlin, Heidelberg: Springer Berlin Heidelberg), 452–473.

Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., et al. (2007). "Provable data possession at untrusted stores," in Proceedings of the 14th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, 28-31 Oct. 2007, 598–609. doi:10.1145/1315245.1315318

Ateniese, G., Di Pietro, R., Mancini, L. V., and Tsudik, G. (2008). "Scalable and efficient provable data possession," in Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, New York, NY, USA, September 22, 2008, 1–10.

Bao, F., Deng, R. H., and Zhu, H. (2003). "Variations of diffie-hellman problem," in Proceedings of the International conference on information and communications security (Springer), Berlin, Heidelberg, October 10, 2003, 301–312.

Boneh, D., Lynn, B., and Shacham, H. (2001). "Short signatures from the weil pairing," in Proceedings of the International conference on the theory and application of cryptology and information security (Springer), Berlin, Heidelberg, November 20, 2001, 514–532.

Chen, X., Li, J., Ma, J., Tang, Q., and Lou, W. (2014). New algorithms for secure outsourcing of modular exponentiations. *IEEE Trans. Parallel Distrib. Syst.* 25, 2386–2396. doi:10.1109/tpds.2013.180

Erway, C. C., Küpçü, A., Papamanthou, C., and Tamassia, R. (2015). Dynamic provable data possession. *ACM Trans. Inf. Syst. Secur.* 17, 1–29. doi:10.1145/2699909

Gao, X., Yu, J., Chang, Y., Wang, H., and Fan, J. (2021). Checking only when it is necessary: Enabling integrity auditing based on the keyword with sensitive information privacy for encrypted cloud data. *IEEE Trans. Dependable Secure Comput.*, 1–1. doi:10.1109/TDSC.2021.3106780

GMP (1991). The gnu multiple precision arithmetic library (gmp). Available at: http://gmplib.org. (Accessed September 1, 2022).

Guo, W., Qin, S., Gao, F., Zhang, H., Li, W., Jin, Z., et al. (2020). Dynamic proof of data possession and replication with tree sharing and batch verification in the cloud. *IEEE Trans. Serv. Comput.* 15, 1813–1824. doi:10.1109/TSC.2020.3022812

He, D., Kumar, N., Zeadally, S., and Wang, H. (2018a). Certificateless provable data possession scheme for cloud-based smart grid data management systems. *IEEE Trans. Ind. Inf.* 14, 1232–1241. doi:10.1109/TII.2017.2761806

He, D., Zeadally, S., and Wu, L. (2018b). Certificateless public auditing scheme for cloud-assisted wireless body area networks. *IEEE Syst. J.* 12, 64–73. doi:10.1109/JSYST.2015.2428620

Ji, Y., Shao, B., Chang, J., Xu, M., and Xue, R. (2022). Identity-based remote data checking with a designated verifier. *J. Cloud Comput. (Heidelb).* 11, 7–14. doi:10.1186/s13677-022-00279-5

Juels, A., and Kaliski, B. S. (2007). "Pors: Proofs of retrievability for large files," in Proceedings of the 14th ACM Conference on Computer and Communications Security, New York, NY, USA, October 28, 2007, 584–597. doi:10.1145/1315245.1315317

Li, B., He, Q., Chen, F., Dai, H., Jin, H., Xiang, Y., et al. (2021). Cooperative assurance of cache data integrity for mobile edge computing. *IEEE Trans. Inf. Forensic. Secur.* 16, 4648–4662. doi:10.1109/tifs.2021.3111747

Li, Y., Yu, Y., Yang, B., Min, G., and Wu, H. (2018). Privacy preserving cloud data auditing with efficient key update. *Future Gener. Comput. Syst.* 78, 789–798. doi:10.1016/j.future.2016.09.003

Liu, Y., Yu, J., Fan, J., Vijayakumar, P., and Chang, V. (2021). Achieving privacy-preserving dsse for intelligent iot healthcare system. *IEEE Trans. Ind. Inf.* 18, 2010–2020. doi:10.1109/tii.2021.3100873

Liu, Z., Ren, L., Li, R., Liu, Q., and Zhao, Y. (2022). Id-based sanitizable signature data integrity auditing scheme with privacy-preserving. *Comput. Secur.* 121, 102858. doi:10.1016/j.cose.2022.102858

Lu, Q., Li, S., Zhang, J., and Jiang, R. (2022). Pedr: Exploiting phase error drift range to detect full-model rogue access point attacks. *Comput. Secur.* 114, 102581. doi:10.1016/j.cose.2021.102581

Lynn, B. (2015). The pairing-based cryptographic library. Available at: https://crypto.stanford.edu/pbc/. (Accessed September 1, 2022).

McCurley, K. S. (1990). The discrete logarithm problem. *Proc. Symp. Appl. Math (USA)* 42, 49–74.

McDaniel, P., and McLaughlin, S. (2009). Security and privacy challenges in the smart grid. *IEEE Secur. Priv. Mag.* 7, 75–77. doi:10.1109/MSP.2009.76

Peng, X., Xian, H., Lu, Q., and Lu, X. (2021). Semantics aware adversarial malware examples generation for black-box attacks. *Appl. Soft Comput.* 109, 107506. doi:10.1016/j.asoc.2021.107506

Shacham, H., and Waters, B. (2008). "Compact proofs of retrievability," in *Advances in cryptology - asiacrypt 2008*. Editor J. Pieprzyk (Berlin, Heidelberg: Springer Berlin Heidelberg), 90–107.

Shamir, A. (1985). "Identity-based cryptosystems and signature schemes," in *Advances in cryptology*. Editors G. R. Blakley, and D. Chaum (Berlin, Heidelberg: Springer Berlin Heidelberg), 47–53.

Shao, Y., Tian, C., Han, L., Xian, H., and Yu, J. (2022). Privacy-preserving and verifiable cloud-aided disease diagnosis and prediction with hyperplane decision-based classifier. *IEEE Internet Things J.*, 9, 21648–21661. doi:10.1109/JIOT.2022.3181734

Shen, W., Qin, J., Yu, J., Hao, R., and Hu, J. (2019). Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage. *IEEE Trans. Inf. Forensic. Secur.* 14, 331–346. doi:10.1109/tifs.2018.2850312

Wang, B., Li, B., Li, H., and Li, F. (2013). "Certificateless public auditing for data integrity in the cloud," in Proceedings of the 2013 IEEE Conference on Communications and Network Security (CNS)., National Harbor, MD, USA, October 14–16, 2013 18, 136–144. doi:10.1109/CNS.2013.6682701

Wang, H., Wang, Q., and He, D. (2019). Blockchain-based private provable data possession. *IEEE Trans. Dependable Secure Comput.*, 1–10. doi:10.1109/TDSC.2019.2949809

Wang, H., Wu, Q., Qin, B., and Domingo Ferrer, J. (2014). Identity-based remote data possession checking in public clouds. *IET Inf. Secur.* 8, 114–121. doi:10.1049/iet-ifs.2012.0271

Wang, Y., Wu, Q., Qin, B., Shi, W., Deng, R. H., and Hu, J. (2017a). Identity-based data outsourcing with comprehensive auditing in clouds. *IEEE Trans. Inf. Forensic. Secur.* 12, 940–952. doi:10.1109/tifs.2016.2646913

Wang, Y., Wu, Q., Qin, B., Tang, S., and Susilo, W. (2017b). Online/offline provable data possession. *IEEE Trans. Inf. Forensic. Secur.* 12, 1182–1194. doi:10.1109/TIFS.2017.2656461

Wu, G., Mu, Y., Susilo, W., Guo, F., and Zhang, F. (2019). Privacy-preserving certificateless cloud auditing with multiple users. *Wirel. Pers. Commun.* 106, 1161–1182. doi:10.1007/s11277-019-06208-1

Xu, Y., Sun, S., Cui, J., and Zhong, H. (2020). Intrusion-resilient public cloud auditing scheme with authenticator update. *Inf. Sci.* 512, 616–628. doi:10.1016/j.ins.2019.09.080

Xu, Z., He, D., Vijayakumar, P., Gupta, B., and Shen, J. (2021). Certificateless public auditing scheme with data privacy and dynamics in group user model of cloud-assisted medical wsns. *IEEE J. Biomed. Health Inf.*, 1–1. doi:10.1109/jbhi.2021.3128775

Yang, H., Su, Y., Qin, J., and Wang, H. (2020). Privacy-preserving outsourced inner product computation on encrypted database. *IEEE Trans. Dependable Secure Comput.* 19, 1. doi:10.1109/tdsc.2020.3001345

Yu, J., Ren, K., and Wang, C. (2016). Enabling cloud storage auditing with verifiable outsourcing of key updates. *IEEE Trans. Inf. Forensic. Secur.* 11, 1362–1375. doi:10.1109/tifs.2016.2528500

Yu, J., and Wang, H. (2017). Strong key-exposure resilient auditing for secure cloud storage. *IEEE Trans. Inf. Forensic. Secur.* 12, 1931–1940. doi:10.1109/tifs.2017.2695449

Zhang, H., Gao, P., Yu, J., Lin, J., and Xiong, N. N. (2021a). Machine learning on cloud with blockchain: A secure, verifiable and fair approach to outsource the linear regression. arXiv preprint arXiv:2101.02334.

Zhang, H., Tong, L., Yu, J., and Lin, J. (2021b). Blockchain-aided privacy-preserving outsourcing algorithms of bilinear pairings for internet of things devices. *IEEE Internet Things J.* 8, 15596–15607. doi:10.1109/jiot.2021.3073500

Zhang J, J., Li, Z., Wang, B., Wang, X. A., and Ogiela, U. (2020). Enhanced certificateless auditing protocols for cloud data management and transformative computation. *Inf. Process. Manag.* 57, 102287. doi:10.1016/j.ipm.2020.102287

Zhang, X., Huang, C., Zhang, Y., and Cao, S. (2021c). Enabling verifiable privacy-preserving multi-type data aggregation in smart grids. *IEEE Trans. Dependable Secure Comput.*, 1–1. doi:10.1109/TDSC.2021.3124546

Zhang, X., Zhao, J., Xu, C., Li, H., Wang, H., and Zhang, Y. (2019). Cipppa: Conditional identity privacy-preserving public auditing for cloud-based wbans against malicious auditors. *IEEE Trans. Cloud Comput.* 9, 1362–1375. doi:10.1109/TCC.2019.2927219

Zhang Y, Y., Yu, J., Hao, R., Wang, C., Ren, K., Jia, X., et al. (2020). Towards identification of molecular mechanism in which the overexpression of wheat cytosolic and plastid glutamine synthetases in tobacco enhanced drought tolerance. *Plant Physiol. biochem.* 17, 608–620. doi:10.1016/j.plaphy.2020.04.013

Zhang, Y., Xu, C., Yu, S., Li, H., and Zhang, X. (2015). Sclpv: Secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors. *IEEE Trans. Comput. Soc. Syst.* 2, 159–170. doi:10.1109/TCSS.2016.2517205

Zhou, L., Fu, A., Mu, Y., Wang, H., Yu, S., and Sun, Y. (2021). Multicopy provable data possession scheme supporting data dynamics for cloud-based electronic medical record system. *Inf. Sci.* 545, 254–276. doi:10.1016/j.ins.2020.08.031

Zhou, L., Fu, A., Yang, G., Wang, H., and Zhang, Y. (2022). Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics. *IEEE Trans. Dependable Secure Comput.* 19, 1–1132. doi:10.1109/TDSC.2020.3013927