Hindawi
Security and Communication Networks
Volume 2020, Article ID 8837456, 10 pages
https://doi.org/10.1155/2020/8837456

*Research Article*

# Attribute-Based User Revocable Data Integrity Audit for Internet-of-Things Devices in Cloud Storage

**Yaowei Wang**[1,2] **Chen Chen**[3] **Zhenwei Chen**[3] **and Jiangyong He**[4]

[1]*School of Telecommunications Engineering, Xidian University, Xi'an 710071, China*
[2]*School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*
[3]*School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*
[4]*National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*

Correspondence should be addressed to Chen Chen; alchen67@163.com

Received 6 August 2020; Revised 16 September 2020; Accepted 5 October 2020; Published 21 October 2020

Academic Editor: Ximeng Liu



Mobile crowdsensing (MCS) is a sensing paradigm exploiting the capabilities of mobile devices (Internet-of-Things devices, smartphones, etc.) to gather large volume of data. MCS has been widely used in cloud storage environment. However, MCS often faces the challenge of data integrity and user revocation issues. To solve these challenges, this paper uses attribute-based revocable signature mechanisms to construct a data integrity auditing scheme for IoT devices in the cloud storage environment. Users use attribute private keys to generate attribute signatures, and limit the user's permission to use shared data through access policy control. Only when the user attribute is included in the global attribute set, and the attribute threshold is not less than the specified number, the user can use the attribute key for the data to generate a valid signature that can be authenticated under the control of the signature strategy. At the same time, the group manager (GM) can send secret information to a third-party auditor (TPA) to track the creator of the signature, to withdraw the user's access to data when the business changes, and realize the safe revocation of user group membership. Formal security analysis and experimental results show that the proposed data-auditing solution is suitable for IoT devices in the cloud storage environment with respect to security and performance.

## 1. Introduction

With the widespread use of MCS systems exemplified by Internet-of-Things (IoT) and mobile communication devices [1], the way users collect and use data has gradually become more diverse. While the portable terminal brings convenience to users' work and life [2–5], it also has certain limitations, such as limited storage space, difficulty in achieving data synchronization between different terminal devices, and the immediacy of accessing data, which makes data cannot be fully utilized. In order to improve user's work efficiency, improve data utilization rate, and reduce local data management and maintenance costs, the cloud storage technology has been promoted.

Because the storage service provider is not completely trusted, the data entrusted by the user to the third-party storage have potential security risks [6–12] such as the deletion of data with low usage rates, the fact that the data were damaged due to attacks is concealed, storage does not meet user requirements, and data are maliciously leaked. Therefore, the proposed data integrity auditing technology can help users to ensure that the integrity and availability of data are not damaged when using incompletely trusted cloud storage services, thereby better monitoring the data storage status. For example, in 2019, Alibaba Cloud went down due to server failure, which led to a large area of paralysis of the APP and website produced by the company that entrusted the software business to Alibaba Cloud, resulting in user business losses. In case of such losses, for solve the problems in time, users hope to get timely problem feedback from the service provider. However, if the service provider deliberately conceals the loss, the user's interests will be damaged.

Therefore, to ensure the security of cloud storage services, one of the urgent problems is to propose more efficient mechanisms to resist security threats.

The use of data is no longer limited to a single user. In many cases, the data will be shared in a specified work area for multiple users to access. For this type of sharing scenario, users also face many security threats, such as abuse of user access rights, malicious collusion between revoked users and storage causes collusion attacks on data, dynamic data modification issues, and user privacy leaks. Adding an effective user authority grant mechanism to the auditing scheme can ensure secure data access from the perspective of users and service providers. While effectively ensuring the security of data, it can also ensure that the legitimate rights and interests of data users are not infringed.

When hackers attack the server, they can directly obtain the data stored in the cloud for illegal transactions. For example, as much as 87 GB of user data stored by MEGA, a cloud storage service provider, has been leaked. According to the amount of data leaked, this data leakage event has become the largest security accident in history. Hackers attacked the servers of MyHeritage and other websites to obtain user information, resulting in up to 617 million private data being sold on the dark Internet. The user password stored in Facebook plaintext is also publicly viewed by the company's employees, and the user's privacy is gone, and the unencrypted data are likely to be directly used to cause user losses. All kinds of examples show that restricting the access rights of users and storing the data on the third-party server after encryption can better protect our key data.

*1.1. Related Work.* With the development of science and technology, the increasingly changing way of work puts forward higher requirements for the function and security of cloud storage services. In addition to solving the data storage problems of users, cloud storage services are expected to meet the needs of users to access data anytime, anywhere. The cloud storage service needs to ensure that the data stored in the server are not modifiable by either the cloud server or the users sharing the data without the user's permission. However, the data stored in the third party cannot be under the supervision of users all the time. On the one hand, it does not meet the actual situation; on the other hand, it wastes too much resources, thus losing the significance of using cloud storage services. Therefore, with the development of cloud storage technology, an efficient and low-cost audit scheme for data integrity of cloud storage has become one of the hot issues in this field.

Today, the data storage will be more flexible with the scale of data access. If we only rely on the DO (data owner) to audit the integrity of data is not conducive to the development and use of cloud storage services, and with the increase of data volume, the audit burden of the DO will be increased, especially for users with limited computing power and resources. At this time, we can reduce the DO burden and improve the audit efficiency by dispersing the audit work. Considering the actual situation, more local cases have proposed a new form of data integrity audit; that is, the audit work is entrusted to the outside, which is called public audit. In 2007, Ateniese et al. proposed an original public audit model based on RSA homomorphism markers [13]. The verifier only needs to store a small amount of raw data to verify the integrity of the data stored on the server. The scheme can realize remote verification of data integrity by random sampling of data blocks, which improves the reliability of audit and reduces the audit burden of users. In 2010, the public audit scheme [14] proposed by Wang et al. is based on homomorphic authenticator and random mask technology, which not only realizes the third-party audit (TPA) batch audit but also ensures that TPA cannot obtain any information about the data in the process of auditing data. In recent years, there is also about the use of blockchain to achieve the audit of Internet-of-Things [15–17] data in the cloud storage environment.

The proxy re-encryption technology [18] used in the scheme proposed by Ateniese et al. can control the authorized decryptor's decryption authority on the ciphertext, so when the business changes, the decryptor's decryption authority on the ciphertext can be recovered, further ensuring that the ciphertext can only be decrypted by the designated user. To achieve secure user revocation, Jiang et al. constructed a new data audit scheme [19] against collusion attack by using the group signature technology with good functions and data-processing mechanism. This scheme solved the potential collusion attack problem in the scheme [20] constructed by Yuan et al. and realized the good attributes of audit disclosure. In the scheme Panda, Wang et al. [21] proposed to let cloud sign data instead of users, supporting batch audit and user revocation. Then, the relevant revocation scheme [22–24] which combines attribute-based encryption (ABE) [25] and proxy re-encryption technology is proposed. We call it revocable attribute-based encryption (R-ABE) here. Sahai et al. used an attribute-based encryption scheme [26] to construct the scheme of user revocation. In this scheme, cryptograph delegation technology and double ABE are used to allow CSP to be responsible for updating cryptograph [27]. The attribute authorization center holds the private key and the update key used for indirect revocation. The validity of the update key is determined by its own effective time, and the update key only performs key update operations for users who are not revoked. However, due to the need to use two ABE schemes in its construction, this kind of revocation method is inefficient and not suitable for a large number of frequently updated application scenarios. As an improvement of this kind of scheme, in scheme [28], to improve the efficiency of CSP update, a new access mechanism is used to replace the time-based update key in the R-ABE scheme, which saves the process of entrusting the key to CSP. The data owner (DO) stores the original ciphertext to the CSP during the revocation period. If there is a ciphertext query beyond this revocation period, CSP will send the ciphertext that takes effect within the current time limit to the legitimate inquirer. As long as the user meets the access policy and revocation time limit specified by the DO, the ciphertext can be decrypted by the user. This scheme combines identity-based encryption (IBE) and

time-encoding mechanism to achieve fine-grained access control and data sharing [29–31]. Inspired by such user revocation schemes, some attribute-based user revocable data integrity audit schemes have been proposed [24].

In 2017, Yu et al. proposed an attribute-based cloud data audit protocol [32] to complete the check of data status while achieving efficient key processing. The data are uploaded to the cloud by the user, but the so-called attributes are needed for the subsequent identification of these data providers by the cloud. This is done by the specifier. Tian et al. ensure the anonymity of users when auditing data integrity artificially, prevent the third party from inferring the identity information of data owners from the inspection program, propose a new concept of cloud data integrity audit based on attributes [33], so as to easily realize the anonymity of users, and propose the security model of such system. Yan et al. proposed a novel remote data-holding test scheme [34]. Based on the original remote data-holding checking (RDPC) protocol, we can prevent forgery attack and realize dynamic data update. Fu et al. proposed a privacy audit scheme NPP [35], which supports the privacy protection of multiple parties in the group and realizes effective user revocation. The new data structure designed based on binary tree can effectively track the change of data. At the same time, the TPA in this scheme needs to obtain the authorization of GM when checking the data of the cloud server, which makes CSP resist the malicious audit request to a certain extent to ensure the effectiveness of CSP work.

Cloud storage service not only provides users with convenient data usage [36] but also makes users unable to master the absolute control of data. Therefore, it is very important to provide an effective CSP supervision mechanism for improving users' trust in cloud services. The integrity verification technology of data has been one of the hot issues in the field of information security since its birth. Considering that in reality, cloud storage service usually has complex application scenarios such as multicloud, remote access, and duplicate data storage, in which user data sharing can help promote the use of CSP. The existing cloud storage data integrity audit scheme with user revocation attribute is suitable for sharing data between remote data access and group users face has strong practicability. Therefore, our main direction in this scheme is user security revocation in data integrity audit scheme.

*1.2. Our Contribution.* In order to achieve an efficient audit of user security while ensuring data integrity, our scheme has the following contributions:

(i) Public audit: a trusted third-party organization is entrusted with strong computing power to monitor the storage status of the data stored by CSP.

(ii) Correctness: it can effectively verify whether the data stored in the cloud server are correctly stored, and can effectively supervise the storage behavior of CSP.

(iii) Access control: only users who meet the specified attribute policy can access the shared data.

(iv) Secure user revocation: the revoked user cannot pass identity verification and cannot access or modify shared data for dynamic operations.

(v) Traceability: if a user abuses data or poses a threat to data security, the user's identity can be identified and the user's true identity can be tracked.

*1.3. Organization.* This paper is organized as follows: Section 1 describes the research background and related work. Section 2 introduces the research content of this paper. Section 3 describes the professional basic knowledge used in this paper. Section 4 elaborates the details of the plan. Section 5 has carried on the security proof to the proposed scheme. In Section 6, the scheme is compared and simulated. Section 7 summarizes the whole work.

## 2. Problem Statement

*2.1. System Model.* As shown in Figure 1, the entities involved in this scheme are cloud storage service provider (CSP), cloud storage service consumer, third-party auditor (TPA) and key generation center (KGC), and a user group with various memberships. The specific functions and definitions of each party are as follows:

(i) CSP: it provides outsourcing data calculation and storage services for user groups, and verifies group members' identity when group members access data. When the user needs to verify the data storage status, data integrity data are generated and sent to the TPA for verification of the data integrity certificate.

(ii) TPA: it verifies the data integrity certificate generated by CSP, saving the user data audit burden.

(iii) Users: it refers to users who need cloud services, by purchasing cloud services, and the data storage work is entrusted to the cloud server for execution.

(iv) KGC: it is responsible for generating parameters for the system and generating attribute keys for all users.

*2.2. Security Model.* The security threats we considered in this scheme mainly are as follows:

(i) Semitrusted CSP: the CSP may not faithfully report the data storage status to the user and maliciously cover up data damage or loss.

(ii) Revoked user: the revoked user colluded with the CSP using the expired signature, concealed the true modification of the data, and conducted a collusion attack on the database.

(iii) Third-party auditor: TPA may be honest but curious. If the privacy of user identity is not anonymously protected, TPA is likely to analyze the user identity and user data information.

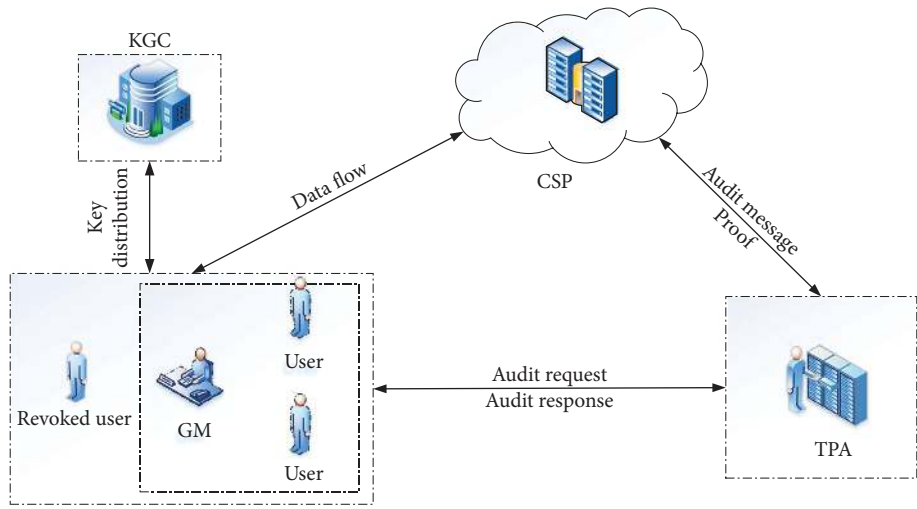


Figure 1: System model.

*2.3. Security Definition.* A game is constructed that exists between the adversary $A$ and the challenger $B$, and the security of the scheme proposed in this chapter is proved through the operation of the game. The details are as follows:

(1) System setup phase: the challenger runs the system setup algorithm, obtains the system public parameters, and sends it to the adversary. It keeps the master key msk and user revocation list.

(2) Query phase: in the query phase, the adversary will perform hash query, key query, signature query, and query of the generated proof.

(i) Hash query (query $H_1$ and query $H_2$, respectively): through these two types of query adversaries, the obtained information such as user attributes and signature policies can be converted into elements on. The challenger separately generates a query list to observe the challenge from the adversary.

(ii) Key query: the identity and related attribute set are input, and the challenger generates the relevant key to send to the $A$.

(iii) Signature query: the $A$ asks about the signature on the message, and runs the algorithm to generate $\sigma$ and send it to $A$.

(iv) Proof query: the data integrity proof is obtained from CSP, and the proof, audit message, and signature are sent to TPA for data integrity verification.

(3) Forgery phase: the adversary outputs a tuple containing elements such as forged signatures and data integrity proof. If it can pass the verification, the game is aborted and the adversary wins.

If $A$ can win the game with negligible advantage $\varepsilon$, then the scheme is security.

## 3. Preliminaries and Complexity Assumption

*3.1. Notations.* The main notations used in this paper are shown and explained in Table 1.

*3.2. Bilinear Groups.* Let $G$ and $G_T$ are two multiplicative cyclic groups of prime order $p$. $e$ is a bilinear [37] map: $G \times G \longrightarrow G_T$ with the following properties:

(i) Bilinearity: for all $u, v \in G$, and $a, b \in Z_p$, $e(u^x, v^y) = e(u, v)^{xy}$.

(ii) Nondegeneracy: $e(g, g) \neq 1$.

(iii) Computability: there is an effective algorithm to compute bilinear maps $e$.

*3.3. Computational Diffie–Hellman Problem.* $G$ is defined as a cyclic group of prime order [38, 39] $g$ is the generator of $G$, given $g^a, g^b \in G, a, b \in_R Z_p^*$, and the probability of $g^{ab}$ calculated by the adversary $A$ in polynomial time is negligible; then, there is $\mathrm{adv}_A^{\mathrm{CDH}} = [A(g^a, g^b) = g^{ab}] \leq \varepsilon$.

*3.4. Revocable Signature Based on Attribute.* In attribute-based signature (ABS), users sign messages using any of their attribute predicates published from the attribute authority. Under this concept, the signature is not to prove the identity of the person signing the message, but to declare the properties owned by the underlying signer. In ABS, even if malicious users collude with each other to synthesize attributes that can generate effective signatures, users cannot forge signatures with attributes that they do not have.

Users get secret key from GM according to their attributes and choose signature strategy that meets the attribute requirements. Through the secret key, users can calculate the data signature based on this signature strategy. The verifier will not get any information about identity or attributes when verifying the user's signature, and just need to verify the attributes to meet the signature policy. In this section, we will describe in detail the four main algorithms of this signature scheme [40] as follows:

(i) $\mathrm{KeyGen}(\mathrm{id}, \Delta, \mathrm{msk}, \mathrm{params}) \longrightarrow (\mathrm{sk}_i, \mathrm{SK}_{\mathrm{id},\Delta}, \mathrm{RL})$: the user identity and attributes are input, the msk and parameters generated during system initialization are input, and the user private key and global

Table 1: Main notations in this paper.

| Notations | Description |
|---|---|
| $\xi_p$ | User's revocation secret value |
| rk | Revocation key |
| msk | Master private key |
| params | Public parameters |
| $\Phi$ | Attribute domain |
| $sk_i$ | User's private key |
| $SK_{is,\Delta}$ | Global private key |
| $list_{id}$ | User information table |
| $\Delta'$ | Minimum authorized set of attributes |
| AM | Audit message |
| $k$ | Attribute threshold |
| $c_i$ | Commitment value |
| $\sigma$ | User's signature |
| $\Lambda$ | Data integrity proof |

private key as well as the user revocation list are output.

(ii) Sign $(M, \Phi, \gamma, SK_{id,\Delta}, params) \longrightarrow (\sigma)$: the user global property collection is set. When the user property belongs to this collection, the data signature is generated according to the user private key.

(iii) Verify$\sigma, m_i, \Delta, \gamma \longrightarrow (1, \perp)$: the validity of the signature is verified. Output verification can be represented by 1 or not by $\perp$.

(iv) Revoke $(list_{id}, rk) \longrightarrow (id, k_{id})$: the revoked user identity is restored, and RLis joined.

*3.5. Threshold Strategy.* Assuming that $(\Delta, \gamma, \Phi)$ is a threshold strategy, let Delta be a set containing $n$ attributes, and the threshold is $\gamma$; then, $\Delta = \{A | A \subseteq \Delta, |A| \leq \gamma\}$, at least having $\gamma$ attributes in the attribute set.

*3.6. Automorphic Signature.* In the signature generation process, the scheme embeds the verification key in the message space, and the data and signature in the message space are considered to be composed of elements in the bilinear group. Such a signature scheme [41, 42] is called automorphic signatures (ASs). The signature validity verification on the message data is verified by a set of paired product equations. The self-constructed signature constructed based on the CDH hypothesis can resist the chosen-message attack (CMA) from adaptive adversaries. We give the general structure of the self-constructed signature scheme as follows:

(i) Setup: it is supposed that there is a quintuple composed of bilinear group elements tuple = $(e, g, G, G_T, p)$. At the same time, $G \longrightarrow (x, y, z)$ is selected, and a data space $D = (W^d, V^d)$ composed of data is defined, where $d \in Z_p$.

(ii) KeyGen: $h \in Z_p$ is selected to calculate the private key $k = g^h$.

(iii) Sign: the data $d_i \in D$ to be signed are input, the random number $s, r \longrightarrow Z_p$ is selected, and the signature is calculated as $\sigma = (\{B = \sum y \cdot x \cdot d_{i1 \leq i \leq n}\}^{1/h+s}, E = g^s, Q = x^s, T = g^r, I = x^r)$.

(iv) Verify: it is verified that the signature $\sigma$ generated in the previous step meets the following three verification equations $e(B, h \cdot E) = e(y \cdot d, g)e(z \cdot g^r), e(E, x) = e(g, Q), e(g^r, x) = e(g, x^r)$ to determine the validity of the value.

## 4. Scheme Construction

*4.1. Scheme Framework.* The construction of this scheme includes setup, key generation, signature, proof generation, verification, user security revocation, and so on. The basic definition of the algorithm is as follows:

Setup $(1^\lambda) \longrightarrow (params, msk, \xi_p)$: the algorithm inputs the security parameter $\lambda$, and the output is the public parameter and the master key of the system. In addition, it generates the secret value $\xi_p$ about the user's revocation. This step is completed by the attribute authorization center.

KeyGen $(id, \Delta, msk, params) \longrightarrow (sk_i, SK_{id,\Delta}, RL)$: in this algorithm, the attribute authorization center takes the user identity id and the associated user attribute set $\Delta$, the system parameters params, and the master key msk which was generated in the system initialization as the algorithm input, outputs the user private key $sk_i$ and the global attribute private key $SK_{id,\Delta}$ after calculation, and stores them together with the list RL used to judge the user revocation.

Sign $(M, \Phi, \gamma, SK_{id,\Delta}, params) \longrightarrow (c_i, \alpha_i, \sigma)$: this algorithm takes data $M$, user attribute domain $\Phi$, attribute threshold $\kappa$, global private key $SK_{id,\Delta}$, and public parameters as input, and then outputs commitment value $c_i$ and corresponding attribute proof $\alpha_i$ and the user's signature $\sigma$ on data through calculation.

Proof $(\sigma, M) \longrightarrow (\Lambda)$: the algorithm inputs data and signature and then generates a data integrity proof $\Lambda$.

Verify $(\Lambda) \longrightarrow (1, \perp)$: it inputs the data integrity proof and verifies the data storage status by equation.

Revoke $(list_{id}, rk) \longrightarrow (id, k_{id})$: taking the identity list $list_{id}$ containing the user's identity information and the revocation key rk as input, the user's real identity can be traced for revocation of the user's identity.

*4.2. A Concrete Scheme.* The main work of this section is to introduce the algorithms in the attribute-based user revocable integrity audit scheme. The details of the algorithms are as follows:

(1) Setup: the attribute authorization center first generates a 5-tuple $\beta = (n, G, G_T, e, g)$ for the system, where $e$ is a bilinear map, $e$: $G \times G \longrightarrow G_T$. Let $G$ and $G_T$ be two bilinear groups. $p$ and $q$ are prime numbers with bit size $\vartheta(\lambda)$ and satisfy the

relationship $n = pq$. Randomly select $\theta \in_R G, \pi \in_R G_q, g \in G, \rho \in Z_n$. Use the hash function $H_1, H_2: 0,1^* \longrightarrow G$, and our scheme can be extended to support any element in $G$. Let the revocation key be $\mathrm{rk} = \xi_p \in Z_n$. This key satisfies the relationship $\xi_p = 0 \bmod q$ and $\xi_p = 1 \bmod p$. Run the automorphic signature generation algorithm to generate the system master key $\mathrm{msk} = (\rho, \mathrm{sk}_{\mathrm{au}})$. The public-private key pair of the signature is $(\mathrm{sk}_{\mathrm{au}}, \mathrm{pk}_{\mathrm{au}})$. The final parameter is $\mathrm{params} = (\beta, g_1, \pi, \pi_1, \theta, H_1, H_2, \mathrm{pk}_{\mathrm{au}}, \Delta)$.

(2) KeyGen: let user attribute be $\mathrm{at}_i$. The attribute set $\Delta$ is contained in the attribute domain $\Phi$. Select the element $\varepsilon_{\mathrm{id}} \in G, r_i \in Z_n$, and then generate a automorphic signature $\sigma_{\varepsilon_{\mathrm{id}}}$. Compute the user's private key as $\mathrm{sk}_i = (\Gamma_i, \Upsilon_i) = (H_1(\mathrm{at}_i)^\rho \varepsilon_{\mathrm{id}}^{r_i}), g^{r_i})$. The global private key is $\mathrm{SK}_{\mathrm{is},\Delta} = (\varepsilon_{\mathrm{id}}, \sigma_{\varepsilon_{\mathrm{id}}}, \mathrm{sk}_{\mathrm{id}\mathrm{at}_i \in \Delta})$. The user identity id and secret value $\varepsilon_{\mathrm{id}}$ are stored in the user information table $\mathrm{list}_{\mathrm{id}}$.

(3) Sign: the user inputs data $M = (m_1, \ldots, m_n)$, attribute set $\Delta$, private key $\mathrm{sk}_i$, and public parameter params. The attribute $\mathrm{at}_i \in \Phi$ owned by the user is set as the minimum authorized set of attributes as $\Delta'$, and the number of matching attributes in $\Delta \cap \Phi$ is set as the threshold $\gamma$. The signature generation follows the following steps:

(i) Choose $z_i \in Z_n$ $\mathrm{at}_i \in \Delta'$ when $x_i = 1$, and $\mathrm{at}_i \notin \Delta'$ when $x_i = 0$. Calculate the commitment value $c_i = (H_1(\mathrm{at}_i)/\theta)^{x_i} \pi^{z_i}$ about $x_i$ and related proof $\omega_i = ((H_1(\mathrm{at}_i)/\theta)^{2x_i - 1} \phi^{z_i})^{z_i}$.

(ii) Set $z = \sum_{\mathrm{at}_i \in \Phi} z_i$, select $t \in_R Z_n$, and compute $H_m = H_2(M, \Phi, \gamma)$and $\sigma_1 = (\prod_{\mathrm{at}_i \in \Delta} \Gamma_i)(H_2(M, \Phi, \gamma))^t \pi_1^z \varepsilon_{\mathrm{id}}^{r'}$, $\sigma_2 = g^t$, $\sigma_3 = \prod_{\mathrm{at}_i \in \Delta'} G_i g^{r'}$.

(iii) Output signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, (c_i, \alpha_i)_{\mathrm{at}_i \in \Delta}, \varepsilon_{id}, \sigma_{\varepsilon_{\mathrm{id}}})$.

(4) Proof: the auditor chooses $i \in I \subseteq [1, n]$ and the random element $k \in Z_q^*$. Output audit message $\mathrm{AM} = \{i, k_i\}_{i \in I}$, and send it to CSP. CSP computes $m' = \sum_{1 \le j \le n} \varepsilon_j m_j \sigma_1' = \prod_{1 \le j \le n} (\sigma_{i,1})^{\varepsilon_j}$, $\sigma_2' = \{\prod_{1 \le j \le n} (\sigma_{i,2})^{\varepsilon_j}\}_{\mathrm{at}_i \in \Delta'} \sigma_3' = \{\prod_{1 \le j \le n} (\sigma_{i,3})^{\varepsilon_j}\}_{\mathrm{at}_i \in \Delta'}$. Output $\Lambda = (m', \sigma_1', \sigma_2', \sigma_3')$.

(5) Verify: the elements in the proof are parsed and input into equation $e(\sigma_1, g) = e(\theta^\gamma \prod_{\mathrm{at}_i \in \Phi} c_i, g_1) e(H_m, \sigma_2) e(\mathrm{com}(\varepsilon_{\mathrm{id}}), \sigma_3) e(h, \alpha_\sigma)$ for verification. Set $\mathrm{com}(\varepsilon_{\mathrm{id}})^{\xi_p} = \varepsilon_{\mathrm{id}}^{\xi_p}$.

(6) Update: if the data change $m \longrightarrow m'$, the signature on the data needs to be recalculated. Update $H_m = H_2(M', \Phi, \gamma)$ to generate a new signature.

(7) Revoke: the attribute authorization center sends the revocation key $\mathrm{rk} = \varepsilon_p$ to a specific user, such as the group manager, to revoke the user's authority, and takes the public parameters, user signature $\sigma$, and the table $\mathrm{list}_{\mathrm{id}}$ corresponding to the secret value $\xi_p$ and user identity as the algorithm input, if $\mathrm{com}(\varepsilon_{\mathrm{id}})^{\xi_p} = \varepsilon_{\mathrm{id}}^{\xi_p}$ is established. This user identity can be successfully tracked. Add the user information to the revocation list RL to realize user revocation. The attribute verification formula is $e(c_i, c_i/(H_i(at_i)/\theta)) = e(\rho, \alpha)$.

## 5. Correctness and Security Analysis

**Theorem 1.** *When the TPA sends an audit message to the cloud server, if the audit response returned by the CSP can be verified by the following equation, it means that the CSP has achieved the correct storage of data.*

*Proof.* The correctness of the storage can be verified by the following equation:

$$
\begin{aligned}
e(\sigma_1, g) &= e\left(\prod_{\mathrm{at}_i \in \Delta} \Gamma_i, g\right) e\left(H_m^t, g\right) \cdot e(H_m, \sigma_2) e \\
&\quad \cdot (\mathrm{com}(\varepsilon_{\mathrm{id}}), \sigma_3) \cdot e(h, \alpha_\sigma) \\
&= e\left(\theta^\gamma \prod_{\mathrm{at}_i \in \Delta_1} \frac{H_1(\mathrm{at}_i^\rho)}{\theta, g}\right) \cdot e\left(H_m, g^t\right) e \\
&\quad \cdot \left(\mathrm{com}(\varepsilon_{\mathrm{id}})^{r'} \mathrm{com}(\varepsilon_{\mathrm{id}})^{r_i}, g\right) e\left(\prod_{\mathrm{at}_i \in \Delta_1} \frac{H_1(\mathrm{at}_i^\rho)}{\theta \cdot \pi^z, g}\right) \\
&= e\left(\theta^\gamma \prod_{\mathrm{at}_i \in \Delta_1} c_i, g\right) e(H_m, \sigma_2) \cdot e(\sigma_3, g) e(\pi_\sigma, h) \\
&= e\left(\theta^\gamma \prod_{\mathrm{at}_i \in \Phi} c_i, g_1\right) \cdot e(H_m, \sigma_2) e \\
&\quad \cdot (\mathrm{com}(\varepsilon_{\mathrm{id}}), \sigma_3) e(h, \alpha_\sigma).
\end{aligned}
\tag{1}
$$

When the equation is established, it shows that CSP has completed the correct storage of data, and the data are stored by the authorized user entrusted to CSP.

**Theorem 2.** *Considering the data security in the attack scenario of choosing message and signature strategy, we use CDH hypothesis to construct data integrity verification scheme, so as to ensure that the adversary cannot pass the legal authentication and damage the data in this scheme with the forged evidence.*

*Proof.* Assuming that there is a polynomial time algorithm $B$, we can solve the CDH problem on $G$ by interacting with the adversary. That is, when there is a generator $\{g_p, g_p^\nu, g_p^\iota\} \in G$, where $g_p$ is the generator of $G$, calculate the value of $g_p^{\nu\iota}$. This shows that the adversary $A$ can successfully forge user signature to obtain data operation authority through authentication.

Game 0: the main stage of this game is the challenge-response parameter generation and correct parameter distribution. Select the generator $h$ in $G$. Randomly

select element $(r_1, \ldots, r_5) \in Z_q$. Let user attribute domain $\Phi$ and automorphic signature public-private key pair $(\mathrm{sk}_{\mathrm{au}}, \mathrm{pk}_{\mathrm{au}})$. Set $g_1 = g_p^{\nu} h^{r_2}$, $g_2 = g_p^{\iota} h^{r_4}$ satisfy the following relationship:

$$\begin{aligned} e(g_1, h) &= e\left(g_p^{\nu} h^{r_2}, h\right) \\ &= e\left(h^{r_1}, h^{r_2/r_1}\right) \\ &= e\left(g_p h^{r_1}\right) \\ &= e(g, h_1). \end{aligned} \tag{2}$$

Finally, send the parameter params to the adversary.

Game 1: in this game stage, the query operation is mainly initiated by $A$.

$Q_{H_1}$: entering the user attribute in the function $H_1$ can convert the value into an element in $G$, so as to facilitate subsequent verification calculations. Take $\mathrm{at}_i$ as input, and run $H_1$ query. The query list $\mathrm{list}_{H_1}$ sends the query response $\{\mathrm{at}_i, c_i, x_i\}$ as output to $A$. The probability of $x_i = 1$ is recorded as $\tau_1$. Compute $H_1(\mathrm{at}_i) = g_2^{c_i}$. When $x_i = 0$, $H_1 = g^{c_i}$ is calculated.

$Q_{H_2}$: the function $H_2$ is used to transform the data and its signature strategy into elements in $G$ to facilitate subsequent verification. Take $(M, \Delta, \gamma)$ as input, and run $H_2$ query. If the item is detected in the query list $\mathrm{list}_{H_2}$, the element $s \in_R Z_n$ is randomly selected, and $(M, \Delta, \gamma, s)$ is output as a query response and sent to $A$. If not, $H_2(M, \Delta, \gamma) = g^s$ is calculated and stored in $\mathrm{list}_{H_2}$ together with $(M, \Delta, \gamma, s)$.

$Q_{\mathrm{key}}$: adversary $A$ wants to get the user's private key generated based on the user's attribute. $A$ queries the key through $B$. When $x_i = 0$, $B$ selects $\mu \in_R Z_n$. When $x_i = 1$, $B$ selects $\mu, \mu' \in_R Z_n^*$ and computes $\varepsilon_{\mathrm{id}} = g^{\mu} g^{\mu'}$. At the same time, $\mathrm{list}_{H_1}$ takes the $\{\mathrm{at}_i, c_i, x_i\}$ item as the output to parse out the commitment information $c_i$. The user private key can be obtained as $\mathrm{sk}_i = ((g_2^{c_i})^{-\mu/\mu'} \gamma_{\mathrm{id}}^{r_i}, g^{r_i} g^{-c_i/\mu'})$.

$Q_{\mathrm{sign}}$: in $B$, after obtaining $\gamma_{\mathrm{id}}$, the user's signature $\sigma$ on the data can be obtained according to the entries queried in $\mathrm{list}_{H_2}$ and $\mathrm{list}_{H_1}$.

$Q_{\mathrm{proof}}$: the adversary selects signature and challenge value to send to $B$ for proof query, and $B$ generates integrity proof $\Lambda = (m', \sigma_1', \sigma_2', \sigma_3')$ sends it to the adversary.

Game 2: the adversary attempts to forge the valid signature of the legitimate user and generate integrity evidence based on the signature. $A$ has tuples $(M', \Delta', \gamma', \sigma')$, where $\sigma'$ is the forged signature of data $M'$ under access policy $(\Delta', \gamma')$. If the signature is valid, then equation $e(c_i, c_i'/(H_{\mathrm{at}_i}/\theta)) = e(\rho, \alpha_i)$ holds. That is, $A$ traverses the list $\mathrm{list}_{H_1}, \mathrm{list}_{H_2}, \mathrm{list}_{H_{\mathrm{key}}}$, obtains $H_1(\mathrm{at}_i) = g_2^{c_i}, H_2(M, \Delta, \gamma) = g^s$, and calculates $g_p^{\mu\iota} = [(\sigma_1' \sigma_2^{-s'} \sigma_3^{-\mu'})^{\delta_p} g_p^{\pi c_i}]^{1/\gamma}$.

We assume that $A$ can compute the probability of $g_p^{\mu\iota}$ as $\mathrm{adv}_{\lambda}^{\mathrm{CDH}} = |\Pr(M, \Delta, \gamma, \sigma) - \Pr(M', \Delta', \gamma', \sigma')| \le \varepsilon$. That is to say, $A$ can break the security with the advantage of $\varepsilon$. In this case, our scheme is secure and can resist signature forgery attacks from adaptive adversaries.

## 6. Performance Analysis

In this section, we compare the computational cost of this paper with other data integrity audit papers [21, 32]. As shown in Table 2, when calculating the cost of each stage of the comparison scheme, in order to make the description more concise and clear, we will use $M$ to represent the multiplication on the multiplicative cyclic group, $P$ to represent the pairing operation, $H$ to represent the hash operation, and $E$ is used to express exponential operation. $r$ represents the number of revoked users, and $n$ represents the number of data blocks. The analysis of the computational cost of each stage in the plan mainly revolves around four operations: multiplication, pairing, hash, and exponent. The experimental environment of this program is a PC with Intel(R) i5-7300HQ CPU@2.5 GHz processor and 8G memory. The Java programming language is used to simulate the algorithm time of the program. The code-writing platform is Eclipse and is based on the Java Pairing Based Cryptography Library (JPBC) library [43] selects a class A elliptic curve for the simulation test of the efficiency of the scheme.

As shown in Figure 2, the main computational overhead in the data integrity proof generation phase comes from the computational storage of the audit message (AM). The computational cost of this scheme at this stage is $2P + 5M + 2H + 3E$. Compared with the other two schemes, it is proved that the cost of generation is related to the size of data block. Our calculation cost at this stage is constant and will not be affected by the size of data. Therefore, it is suitable for large-scale proof generation, greatly reducing the cost of proof.

As shown in Figure 3, here we use $r$ to represent the number of users who have been revoked. Under the assumption that the number of users is $r$, the user revocation time is tested. Since scheme [32] does not include user revocation function, our comparison in revocation phase is only compared with scheme [21]. The computational cost of revoking a single user's operation is constant, but when the number of users increases, the efficiency of this scheme is significantly higher than that of scheme [21]. At this stage, our calculation cost can be recorded as $r(E + 2P)$.

As shown in Figure 4, the cost of the phase in the verification proof does not change with the number of data blocks in the data sequence, and the verification time in this

Table 2: Functionality comparison.

| Scheme | Yu et al. [32] | Wang et al. [21] | Our scheme |
|---|---|---|---|
| ProofGen | $(3+n)M+H$ | $n(M+H)$ | $2P+5M+2H+3E$ |
| Verify | $3P+M+E$ | $3E+2M+4P+2H$ | $H+5P$ |
| Revoke | — | $2r(P+H+M+2E)$ | $r(E+2P)$ |

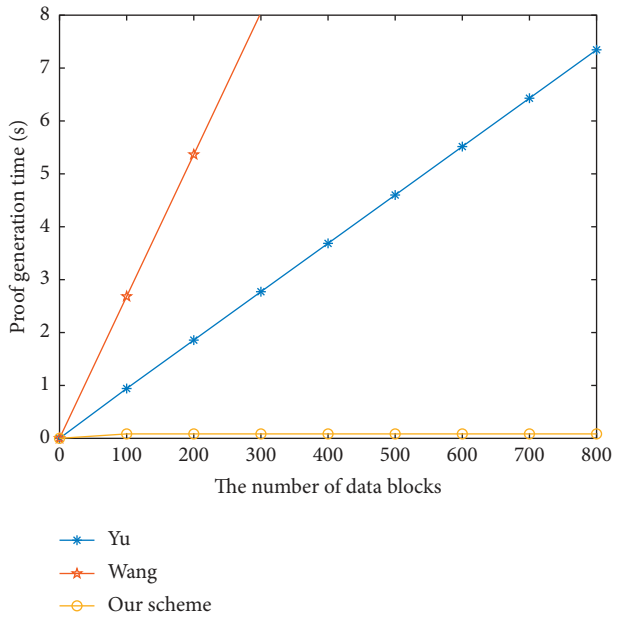


Figure 2: Proof generation time.

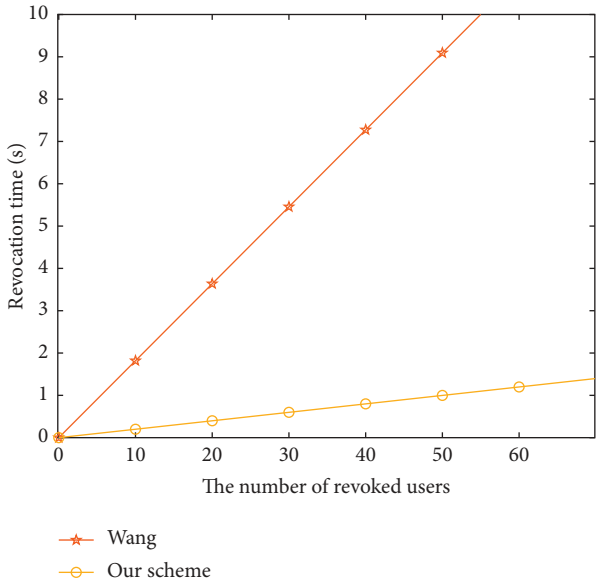


Figure 3: Revocation time.

stage is constant. Compared with the scheme [21, 32], the efficiency of our scheme has also been improved in the data integrity verification stage. The calculation cost of this scheme at this stage can be recorded as $H+5P$.

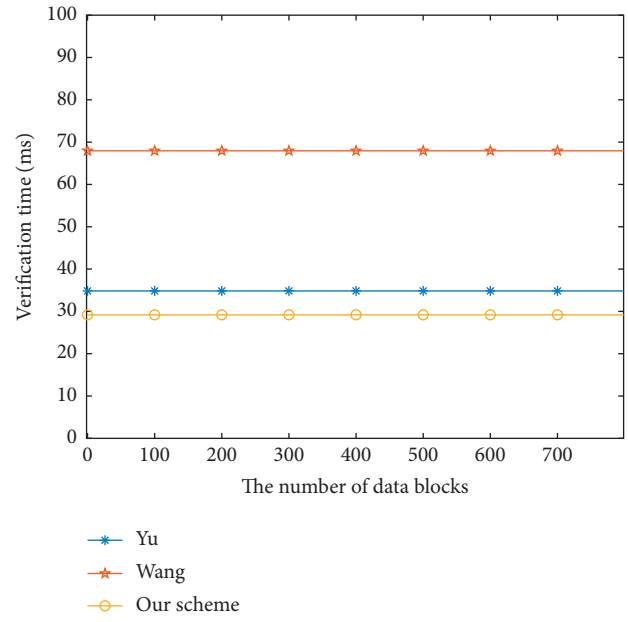


Figure 4: Verification time.

## 7. Conclusion and Future Work

The main discussion in this paper is a user revocable attribute-based data integrity audit scheme. Compared with the scheme of completely anonymous user identity, this scheme can break the anonymity of user signature when necessary, and can be applied to the place where users do not want to be completely anonymous, and the scheme has the function of public audit. In addition, this scheme uses attribute-based signature to realize flexible access permission-granting mechanism, and realizes the unforgeability of signature to resist collusion attack from revoked users.

As the demand for cloud storage services becomes more and more diverse, more and more data security problems are exposed, so we propose the following research directions as the next research content.

*The Authorization Verification of TPA*. In the process of data integrity audit, users entrust a third party to handle the data verification. After receiving the audit challenge from the TPA, the CSP sends a response to send the calculated data certificate to the TPA, but if the application of the TPA is not authorized, it will cause a waste of CSP resources. The introduction of the third-party audit saves the extra audit cost of users and realizes the efficiency of the audit work with its own more professional ability. However, in order to prevent the CSP server from being attacked by DDOS initiated by

malicious TPA, we need to consider an audit authorization mechanism of TPA to limit its audit application.

*Data Batch Audit*. In practice, there are multiple user groups using CSP services at the same time. When multiple user groups send audit requests to the same TPA, TPA needs to have the ability to process audit requests in batches. The solution of this problem can help users enhance their confidence in the reliability of cloud service applications and help developers better promote cloud computing services. So, the problem of batch processing of data integrity audit request in cloud storage environment also needs further research.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors have declared that no conflicts of interest exist.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (grant nos. 62072369 and 62072371), the Innovation Capability Support Program of Shaanxi (grant no. 2020KJXX-052), the Key Research and Development Program of Shaanxi (grant nos. 2019KW-053 and 2020ZDLGY08-04), and the Natural Science Basic Research Plan in Shaanxi Province of China (grant no. 2019JQ-866).

## References

[1] Y. Zhang, R. Deng, E. Bertino, and D. Zheng, "Robust and universal seamless handover authentication in 5G hetnets," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[2] S. E. Mensch and L. Wilkie, "Cell phone security," *International Journal of Strategic Information Technology and Applications*, vol. 9, no. 3, pp. 15–31, 2018.

[3] S. Mensch, "Cell phone security: usage trends and awareness of security issues," in *Proceedings of the International Academic Conferences*, Rome, Italy, November 2016.

[4] A. J. Nicholson, M. D. Corner, and B. D. Noble, "Mobile device security using transient authentication," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1489–1502, 2006.

[5] M. A. Harris and K. P. Patten, "Mobile device security considerations for small- and medium-sized enterprise business mobility," *Information Management & Computer Security*, vol. 22, no. 1, pp. 97–114, 2014.

[6] X. Zhang, H.-t. Du, J.-q. Chen, Y. Lin, and L.-j. Zeng, "Ensure data security in cloud storage," in *Proceedings of the 2011 International Conference on Network Computing and Information Security*, IEEE, Guilin, China, pp. 284–287, May 2011.

[7] R. V. Rao and K. Selvamani, "Data security challenges and its solutions in cloud computing," *Procedia Computer Science*, vol. 48, pp. 204–209, 2015.

[8] L. M. Kaufman, "Data security in the world of cloud computing," *IEEE Security & Privacy*, vol. 7, no. 4, pp. 61–64, 2009.

[9] P. Dinadayalan, S. Jegadeeswari, and D. Gnanambigai, "Data security issues in cloud environment and solutions," in *Proceedings of the 2014 World Congress on Computing and Communication Technologies*, pp. 88–91, IEEE, Trichirappalli, India, February 2014.

[10] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing," in *Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering*, pp. 647–651, IEEE, Hangzhou, China, March 2012.

[11] A. Sangroya, S. Kumar, J. Dhok, and V. Varma, "Towards analyzing data security risks in cloud computing environments," in *International Conference on Information Systems, Technology and Management*, pp. 255–265, Springer, Berlin, Germany, 2010.

[12] Y. Zhang, D. Zheng, and R. H. 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.

[13] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 598–609, Alexandria, VA, USA, November 2007.

[14] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the 14th 2010 IEEE INFOCOM*, IEEE, San Diego, CA, USA, pp. 1–9, March 2010.

[15] C. Machado and A. A. M. Fröhlich, "IOT data integrity verification for cyber-physical systems using blockchain," in *Proceedings of the 2018 IEEE 21st International Symposium on Real-Time Distributed Computing*, pp. 83–90, IEEE, Singapore, May 2018.

[16] Y. Zhang, R. H. 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.

[17] 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*, vol. 5, no. 1, pp. 1939–1374, 2018.

[18] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[19] 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, 2015.

[20] J. Yuan and S. Yu, "Efficient public integrity checking for cloud data sharing with multi-user modification," in *Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications*, pp. 2121–2129, Toronto, Canada, April 2014.

[21] 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.

[22] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proceedings of the 2010 IEEE INFOCOM*, pp. 1–9, IEEE, San Diego, CA, USA, March 2010.

[23] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," in *Proceedings of the 8th ACM SIGSAC Symposium*

*on Information, Computer and Communications Security*, pp. 523–528, Hangzhou, China, May 2013.

[24] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2012.

[25] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and 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.

[26] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," *Lecture Notes in Computer Science*, vol. 7417, pp. 199–217, 2012.

[27] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, "Auditable $\sigma$-time outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 94–105, 2018.

[28] S. Xu, G. Yang, Y. Mu, and R. H. Deng, "Secure fine-grained access control and data sharing for dynamic groups in the cloud," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2101–2113, 2018.

[29] S. Xu, G. Yang, Y. Mu, and X. Liu, "A secure IOT cloud storage system with fine-grained access control and decryption key exposure resistance," *Future Generation Computer Systems*, vol. 97, pp. 284–294, 2019.

[30] S. Xu, Y. Li, R. Deng, Y. Zhang, X. Luo, and X. Liu, "Lightweight and expressive fine-grained access control for healthcare internet-of-things," *IEEE Transactions on Cloud Computing*, 2019.

[31] S. Xu, J. Ning, Y. Li et al., "Match in my way: fine-grained bilateral access control for secure cloud-fog computing," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[32] Y. Yu, Y. Li, B. Yang, W. Susilo, G. Yang, and J. Bai, "Attribute-based cloud data integrity auditing for secure outsourced storage," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 2, pp. 377–390, 2020.

[33] M. Tian, L. Wang, H. Zhong, and J. Chen, "Attribute-based data integrity checking for cloud storage," *Fundamenta Informaticae*, vol. 163, no. 4, pp. 395–411, 2018.

[34] H. Yan, J. Li, J. Han, and Y. Zhang, "A novel efficient remote data possession checking protocol in cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 78–88, 2016.

[35] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, "NPP: a new privacy-aware public auditing scheme for cloud data sharing with group users," *IEEE Transactions on Big Data*, 2017.

[36] J. Ning, X. Huang, W. Susilo, K. Liang, X. Liu, and Y. Zhang, "Dual access control for cloud-based data storage and sharing," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[37] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," *Lecture Notes in Computer Science in IACR Cryptology ePrint Archive*, Springer, Berlin, Germany, pp. 416–432, 2003.

[38] J. H. Cheon and D. H. Lee, "Diffie-hellman problems and bilinear maps," *IACR Cryptology ePrint Archive*, vol. 117, 2002.

[39] A. Joux and K. Nguyen, "Separating decision diffie-hellman from computational diffie-hellman in cryptographic groups," *Journal of Cryptology*, vol. 16, no. 4, pp. 239–247, 2003.

[40] A. Escala, J. Herranz, and P. Morillo, "Revocable attribute-based signatures with adaptive security in the standard model," in *International Conference on Cryptology in Africa*, pp. 224–241, Springer, Berlin, Germany, 2011.

[41] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo, "Structure-preserving signatures and commitments to group elements," in *Annual Cryptology Conference*, pp. 209–236, Springer, Berlin, Germany, 2010.

[42] G. Fuchsbauer, "Automorphic signatures in bilinear groups and an application to round-optimal blind signatures," IACR Cryptology ePrint Archive, p. 320, 2009, http://eprint.iacr.org/.

[43] A. De Caro and V. Iovino, "JPBC: java pairing based cryptography," in *Proceedings of the 2011 IEEE Symposium on Computers and Communications*, pp. 850–855, IEEE, Kerkyra, Greece, June 2011.