A Survey of Public Auditing for Shared Data Storage with User Revocation in Cloud Computing

Chi-Wei Liu[1], Wei-Fu Hsien[2], Chou-Chen Yang[2], and Min-Shiang Hwang[1,3]
(Corresponding author: Min-Shiang Hwang)

Department of Computer Science and Information Engineering, Asia University[1]
No. 500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan (R.O.C.)
(Email: mshwang@asia.edu.tw)
Department of Management Information System, National Chung Hsing University[2]
Department of Medical Research, China Medical University Hospital, China Medical University[3]
No. 91, Hsueh-Shih Road, Taichung 40402, Taiwan (R.O.C.)
(Received May 25, 2015; revised and accepted July 13 & Aug. 4, 2015)

Abstract

Cloud computing technology has matured, so cloud computing produces a wide range of cloud service. Cloud storage services are one of cloud services where cloud service provider can provide storage space to customers. Because cloud storage services bring a lot of convenience, many enterprises and users store the data to the cloud storage. However, the user will outsource data to the cloud storage service, but the user is difficult to manage remote data in the cloud. Therefore, how users verify data integrity is a major challenge. In recent years, public audit is used to verify data integrity by which the user allows other to verify the user's data. Because the feature of cloud service allows users to communicate with each other on the cloud platform, the cloud storage service allows the data owner to share their data to other users. Therefore, public auditing extends to the share data, so the original operation becomes not the same including signature, public audits, dynamic data and user revocation which generates on the situation of shared data. In the paper, we define the requirements of public auditing with shared data and explain four representative approaches which include analysis function, security, and performance requirements. Finally, we provide some topics for future research.

Keywords: Cloud computing, public auditing, share data, user revocation

1 Introduction

Cloud computing is a computing technology, and the internet has grown in recent years. It can share the software and hardware resource, and provide resources to a user's computer or mobile device. The user can obtain a more efficient service because cloud computing can integrate resources. Therefore, in order to achieve cloud computing technology, it must satisfy five basic features: On-demand self-service, Broad network access, Resource pooling, Rapid elasticity and Measured service [40]. However, it is very difficult for general users or small and medium enterprises to construct could environment because they cannot afford the huge costs. Therefore, many information technology companies are finding business opportunities in cloud services. Thus, cloud service providers have joined to build could environments to provide services to the user. Cloud service providers offer three services including Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The cost for users to rent cloud service is cheaper than the cost for users to build cloud environment [1].

Cloud storage service is the most common and popular service among many cloud services (e.g. Google Drive, Dropbox, Amazon S3 and Microsoft OneDrive) for general users. However, users have a bottleneck on the local side storage space because a user needs a large storage space to store a huge amount of data on the situation. Cloud storage service has high capacity and high computation that solve users' difficult problems. Moreover, a user builds a larger storage device which is more expensive than rented cloud storage service. Besides, the user can pay the cloud server provider based on the amount of usage. Then, because cloud storage service provides to access cloud services from web service or applications that utilize the application programming interface (API) by mobile devices (e.g. laptop, table computer, and smart phones), it is convenient to use by users, and it achieves

an ubiquitous service.

Although a cloud storage services has many advantages, it brings a lot of challenging issues which include efficacy and security [26, 34, 38, 46, 47, 48, 63]. One of the big challenges is verifying the integrity of the data because users cannot know how the cloud storage service handles their data. These cloud storage services are provided by commercial enterprises, so it cannot be fully trusted by users. Therefore, the cloud service provider may hide data loss and data errors in the service because of their benefits. However, it is very serious that a user stores data in an untrusted cloud storage. For example, the traditional approach is to download the entire data from the cloud, and then verify data integrity by checking the correctness of digital signatures or hash values of the entire data. Surely, this simple approach is able to check users' data integrity in cloud. However, this is not efficient in the conventional approach because the user spends a lot of resources of communication, computation and storage. Besides, due to a large size of outsourced data and a user's limited resource capability, a user has to find an efficient way to achieve integrity verifications without the local copy of data files.

In order to solve the problem of data integrity verification in the cloud storage service, many studies present different methods and security models [2, 3, 4, 16, 19, 24, 32, 61, 62]. Sookhak et al. [47] surveyed remote data auditing in cloud computing and classified three methods in the following.

First method is named probable data possession (PDP) by Ateniese et al. [2]. They utilized the RSA-based Homomorphic verifiable tag (HVT) to verify the integrity of data storage in the cloud without retrieving the entire data. However, the PDP cannot support to change these stored data which is named static PDP model included [2, 20, 25, 27]. To support dynamic data update in the cloud, Ateniese et al [4] applied symmetric-key cryptography to scalable PDP which is named dynamic PDP included [4, 23, 53]. Wang et al. [50] considered data privacy when TPA verified user's data. TPA can piece together authentication of users' data because TPA can verify users' data on process of public auditing. Thus, it creates data privacy issues. Wang et al. [50] utilized a random mask technology to design an improved approach which can avoid TPA learning users' data which is named privacy-preserving PDP included [62, 33, 50, 54]. Robust PDP including Ateniese et al. [3] utilized a spot-checking mechanism to detect a part of the data corruption, Ateniese et al. [3] utilized forward error checking (FRC) to enhance the arbitrary amount of data corruption and B. Chen and Curtmola [19] utilized a robust dynamic PDP to support error detection of dynamic data update. Second method is named proof of retrievability (POR) by Juels and Kaliski [32]. They embedded the special blocks (named sentinels) to the data and checked the correctness of the sentinels to achieve POR. However, the POR only suits static data storage because dynamic data effects the position of the sentinels. Static POR includes [32, 43, 58]. Cash et al. overcame the difficult problem and improve a dynamic POR [16]. Zheng and Xu proposed a fair and dynamic POR on the 2-3 range tree structure [61]. Third method is named proof of ownership (POW) which considers data deduplication to improve efficient data storage and include [24, 29, 45, 60].

In these studies, the role of the verifier can fall into two categories: privacy verification and public verification. Private verification implies the data owner directly verifying data in the cloud storage service is an efficient way. Public verification implies the data owner allowing other to verify the data owner's data is inefficient because it needs to delegate other verifier by the data owner. In general, a user may have a lot of data files which are stored in cloud storage service. However, a user cannot frequently verify he/she data because it will consume his/her resources so not to process other action. In order to achieve an efficient verification of data integrity, Wang et al. [53] proposed a public auditing scheme where a user can delegate a third party auditor (TPA) to assist the validation reduction to consume his/her computing resources. Then, there are related research bases on Wang et al.'s scheme [53]. Zhu et al. [56] designed another public auditing scheme which can support dynamic data update. With more and more data, it brings new challenges in data integrity. Public auditing for big data storage in the cloud will bring new challenges [18, 38]. Liu et al [36] proposed an efficient verification of fine-grained data update scheme which can support public auditing of big data storage. However, to enhance user's data reliability and availability in the cloud, the cloud server will backup copy user's data. When the user updates data, there backup copy also needs to update. Liu et al. [37] considered cloud server efficiently updates multiple replicas and enhances data availability in the cloud.

Cloud storage service can not only store data but also share information with other users in a group. However, these studies [21, 22, 31, 35, 36, 37, 50, 51, 52, 53, 56] do not consider another advantage of cloud where a user can share data with another user on cloud storage service. Users can share data in the cloud because the cloud platform provides communication between users and others. Therefore, it is very convenient a user wants to share data with another user because this need not be transferred to another user data after downloading. However, users share data in the cloud storage service which still has a problem on the data integrity. Therefore, many recent studies extend public auditing for shared data in the cloud [8, 9, 10, 11, 12, 13, 15, 28, 30, 49, 57, 59]. Because the data is shared with multiple users, it needs to consider dynamic data update of multiple users. When the shared user modifies the shared data block, the shared user need to sign the data block. For example, the user A shares own data with other users (like the user B) in the cloud storage service, and the data is divided into several parts of data blocks which are signed by the user A. The user A allows the user B modifying the user A's data block but the user B has to sign the modified data block. When the

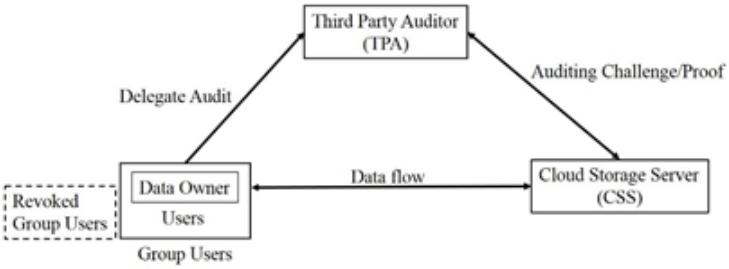

Figure 1: Public auditing with shared data in cloud data storage architecture

user B modifies the shared data block, the user B needs to use his/her private key to sign the modified block. In order to correct the integrity of audit data, the TPA needs to select the corresponding public key to verify the data block (e.g., a data block was signed by user A and it is only correctly verified by user A's public key). Therefore, in the public auditing phase, it is still a problem of identity privacy. However, when a user is revoked from the group because of his/her malicious behavior. The shared data block is signed by the revoked user which needs to be re-signed by the exist user of the group. Therefore, public auditing for shared data has a lot of studies and the system architecture is shown in Figure 1. In the Section 3, we will detail representative approaches [8, 11, 28, 59].

1.1 Requirements

According to [8, 11, 28, 50, 53, 59] studies, they provide the basic requirements of function, security and performance. In our paper, we classify and describe these requirements. Then we use these requirements to analyze the existing scheme in Section 4.

Functional evaluation.

1) Blockless Verification: the auditor can verify data blocks, and needs not to retrieve all audited data blocks in the cloud storage service.
2) Stateless Verification: the auditor need not maintain and update data situation because data situation is maintained by the client and cloud storage service together.
3) Batch Auditing: the auditor can verify the data of different clients at the same time because the auditor can be delegated by a lot of clients.
4) Dynamic Data: the data owner can insert, modify and delete data blocks in the cloud storage service because their data can be continuously updated at any time.
5) Anonymity: the auditor cannot distinguish the identity of the signer on each block during the process of public auditing.
6) Privacy Presenting: the auditor cannot get knowledge to delegate data from the response of the cloud storage service.
7) User Revocation: a user is revoked from the group before an existing user can generate a valid signature on shared data of the blocks signed by the revoked user. The revoked user cannot re-compute valid signatures on shared data.

Security attack evaluation. We list some common attack model and they can analyze whether public auditing scheme can resist the malicious attacker [28, 39].

1) Inside attack: the insiders of the cloud service provider have permission to obtain the client's data in the cloud storage, and take these data to exchange benefits.
2) Forge attack: the cloud server can forge the data tag of data block and deceive the third party auditor.
3) Replace attack: the server can choose another valid pair of data block and data tag (m_i, σ_i) to replace the challenged request (m_j, σ_j), when it already discarded data block or data tag m_i or σ_i.
4) Impersonation attack: an adversary obtains authenticated information of the data owner and cloud storage service and forges another message to pass the verification. Then, the adversary fakes a legal client or cloud storage service and cheats other side.
5) Collusion attack: a revoked user can collude with the malicious cloud server to change the group's shared data.

Performance evaluation.

1) Computing cost: In order to achieve an efficient public auditing, we will analyze the client, TPA and cloud storage service cost on the computing resources.
2) Storage cost: Because the client will upload data to the cloud storage service without the local copy of data files, we will analyze the client, TPA and cloud storage service cost on the storage spaces.

1.2 Contribution

Our contribution can be summarized as the following three aspects: First, we survey the previous researches of public auditing for shared data in the cloud. Then, our paper collect and explain basic requirements in the mechanism. Second, we propose four representative approaches and analyze these approaches by our collected requirements. Third, we summarize the conclusion from the analysis and propose research direction in future work.

1.3 Organization

The rest of paper is organized as follows: In Section 2, we review the related work of public auditability. We discuss the representative approaches of public auditability in detail in Section 3. In Section 4, we analyze the basic requirement in the representative approaches. Finally, we summarize and discuss the future work in Section 5.

2 Related Work

Public integrity auditing with dynamic data for outsourced data storage has caused related research [21, 22, 31, 36, 37, 52, 56]. Wang et al. [52, 53] first proposed an enabling public auditability and data dynamics in the cloud scheme. Their scheme improves data block inserted operation of dynamic data because the inserted operation affects the entire data block which has been sorted. Therefore, they utilized the Merkle Hash Tree (MHT) [41] data structure and bilinear aggregate signature [6] to address dynamic data which can support dynamic index of data block. They extended their scheme to support batch auditing which can improve efficiency. Wang et al. [21, 51] proposed a challenge-response protocol which can determine the data correctness and locate possible errors. However, their scheme only supports partially dynamic data operation. Wang et al. [21] extended [51] to support privacy-preserving third part auditing and correctness analysis of proposed storage verification design. Wang et al. [22, 50] pointed out that Wang et al.'s scheme [53] has data privacy issues which imply the TPA can get the client's data information. Therefore, they use a random mask technology to avoid TPA learning knowledge on every verification process. Wang et al. [22] extended [50] to support dynamic data and prove a secure zero-knowledge leakage public auditing scheme.

Zhu et al. [56] proposed a dynamic audit services for outsourced storage in clouds. They utilized the fragment structure to reduce the storage of signatures, utilized index hash tables to provide the service of dynamic data operation and utilized periodic sampling audit to enhance data integrity. Li et al. [31, 35] considered that the client's resource-constrained device is simple and lightweight. Therefore, they proposed a scheme which a client can delegate TPA to execute high computing process and solve the client's bottleneck before the client uploads data to cloud server. Li et al. [35] extended [31] to improve the users will need to compute the tags for the outsourced data. Liu et al. [36] thought that previous studies are not efficient in dynamic data update because it is a fixed-size block update. Therefore, they proposed a scheme which can support variable-size blocks in dynamic data update and enhance verification efficiency. Liu et al. [37] considered data reliability and availability in the cloud. Consequently, the cloud server will store multiple replicas to enhance data reliability. However, when the stored data is frequently updated, each dynamic update will affect every replica. Therefore, they proposed a multi-replica Merkle hash tree (MR-MHT) to construct replica sub-tree which can enhance data availability in the dynamic data phase. Then, their scheme can support public auditing.

To enhance the previous works [35, 36, 37, 50, 51, 53, 56], there are studies [9, 10, 12, 13, 14, 28, 30, 49, 57] focused on public auditing with shared data in the cloud. Wang et al. [49] proposed a named Knox scheme which is able to audit the integrity of shared data in the cloud for a large group. Unfortunately, their scheme cannot support public auditing. Wang et al. [8, 9] improved drawback of the Knox scheme which implies public auditing so they designed a named Oruta scheme to support public auditing for shared data integrity. Because their scheme utilized ring signature to protect the privacy of users, it did not support a dynamic group. To achieve user revocation, Wang et al. [10, 11] designed a named Panda scheme which is able to audit the integrity of shared data with user revocation in the cloud. They utilized proxy resignature to update the signed data by the revoked user. To preserve the identity of the signer on each block during public auditing, Wang et al. [13, 15] proposed the user of the group to share a global private key. Then, each user can sign blocks by this global private key. However, when a user of the group is compromised or revoked, a global private key has to be re-generated and shared with the existence of the group which will need huge overheard on key management and key distribution. Wang et al. [12] utilized a certificateless scheme to design the first certificateless public auditing mechanism. Their scheme can reduce security risk in certificate management. Wang et al. [14] utilized a multi-signature scheme to design the first multi-owner public auditing mechanism. For example, the correctness of an official document stored in the cloud is confirmed by all the related members before the official document can be announced. Therefore, their scheme can have efficient multi-signature and verify multi-owner data.

Yuan and Yu [58] designed a polynomial commitment scheme which is able to reduce the communication overhead of verification. Yuan and Yu [30, 59] utilized [58] to design a public integrity auditing scheme with multi-user modification. Their scheme uses polynomial authentication tags and proxy tag update techniques, which support public verification and user revocation. Yuan and Yu [59] extended [30] to prevent a compromise attack where single cloud is internal errors or outside attack when cloud server update the authentication tag from the revoked users. Jiang et al. [28] considered the ciphertext store and efficient user revocation where the data owner cannot take part in a user revocation phase. They prevent malicious operation when the cloud server colludes with the revoked user. In Section 3, we will describe these representative approaches in detail.

Table 1: Notations

Notation	Significance
G_1, G_2, G_T	A multiplicative cyclic group
e	A bilinear map $e : G_1 \times G_2 \to G_T$
g	A generator of group G_1
p	The prime order of group G_1
q	A much smaller prime than p
H_1	A hash function $H_1 : \{0,1\}^* \to G_1$
H_2	A hash function $H_2 : \{0,1\}^* \to Z_p$
H_3	A hash function $H_3 : G_1 \to Z_p$
Ψ	A computable isomorphism $\Psi : G_2 \to G_1$ (e.g. $\Psi(g_2) = g_1$
M	The shared data that will be split into n blocks
m_i	A data block of the shared data and will be split into k elements
k	A block element of the shared data block m_i
d	The total number of users in the group
u_i	ith user of the group
sk_i	The user u_i's private key
pk_i	The user u_i's public key
σ_i	Authentication tag generated for shared data block m_i

3 Representative Approaches

Before introducing representative approaches, we list all notation (as shown in Table 1) using in this paper.

3.1 Wang et al.'s Scheme

Wang et al. [8] was the first to propose the scheme which can support shared data on public auditing at the same time because previous studies only considered a data is used by a single user. Therefore, they proposed a privacy-preserving public auditing mechanism for shared data in cloud by the mechanism of one ring to rule them all (Oruta).

They utilized the concept of ring signature [42] to construct homomorphic authenticators, so TPA is able to verify the integrity of shared data. It can achieve efficient verification without retrieving the entire data. They utilized randomly masking technology from C. Wang et al.'s scheme [50] to protect data privacy from public auditing. Meanwhile, they also utilize index hash tables (IHT) from Zhu et al.'s scheme [56] to support dynamic data. Finally, they extend their mechanism to support batch auditing which can allow public auditing on different users simultaneously and improve the efficiency of verification for multiple auditing tasks.

Next we will describe their schemes including setup, public auditing, dynamic data and user revocation phase. Before executing each phase, the CSS need to generate the global parameters: (e, Ψ, p, q, G_1, G_2, G_T, g_1, g_2, H_1, H_2, H_3, d, n, k). Their scheme is as follows:

Setup phase. In the phase, we will describe key generation and signature.

Step 1: A user u_i randomly chooses $x_i \in Z_p$ and computes $w_i = g_2^{x_i}$. Then the user's public key is $pk_i = w_i$ and private key is $sk_i = x_i$. If the user is original, he/she will randomly generate a public aggregate key $pak = (\eta_1, \cdots, \eta_k)$ where η_l are random elements of G_1.

Step 2: The user u_s chooses a block $m_j = (m_{j,1}, \cdots, m_{j,k})$ and the block identifier id_j. The user uses pak to computes $\beta_j = H_1(id_j)\prod_{l=1}^{k}\eta_l^{m_{j.l}} \in G_1$. Then, the user randomly chooses $a_{i,j} \in Z_p$ and gives all the d group members' public keys $(pk_1, \cdots, pk_d) = (w_1, \cdots, w_d)$, and uses a private key sk_s computes a ring signature of this block $\sigma_{j,s} = (\frac{\beta_j}{\Psi(\prod_{i\neq s} w_i^{a_{j,i}})})^{1/x^s} \in G_1$. Therefore, the ring signature of block m_j is $\sigma_j = (\sigma_{j,1}, \cdots, \sigma_{j,d})$.

Public auditing phase. In the phase, we will describe challenge request, proof generation and proof verification.

Step 1: The TPA selects c elements as a subset J of set $[1, n]$ chooses a random value $y_j \in Z_q$, for $j \in J$. Then, the TPA sends the challenged message $\{j, y_j\}_{j\in J}$ to the CSS.

Step 2: The CSS chooses a random value $\tau_i \in Z_q$ and computes $\lambda_l = \eta_l^{\tau_l} \in G_1$, for $l \in [1, k]$. Then, the CSS computes $\mu_l = \sum_{j\in J} y_j m_{j,l} + \tau_l H_3(\lambda_l) in Z_p$, for $l \in [1, k]$. Finally, the CSS aggregates signature as $\phi_i = \prod_{j\in J}\sigma_{j,i}^{y_j}$, for $iin[1, d]$ before the CSS return an auditing proof $\{\{\lambda_1, \cdots, \lambda_k\}, \{\mu_1, \cdots, \mu_k\}, \{\phi_1, \cdots, \phi_d\}, \{id_j\}_{j\in J}\}$.

Index	Block	Virtual index	Random value
$id_j = \{v_j, r_j\}$	m_j	$v_j = j \cdot \delta$	$r_j = H_2(m_j \| v_j)$

Figure 2: Wang et al.'s index hash table (IHT) on Oruta's mechanism

Step 3: The TPA uses public aggregate key $pak = (\eta_1, \cdots, \eta_k)$ and all the group members' public keys $(pk_1, \cdots, pk_d) = (w_1, \cdots, w_d)$ to check the correctness of the auditing proof by computing

$$e(\prod_{j \in J} H_1(id_j)^{y_j} \prod_{l=1}^{k} \eta_l^{\mu_l}, g_2) \stackrel{?}{=} (\prod_{i=1}^{d} e(\phi_i, w_i)) e(\prod_{l=1}^{k} \lambda_l^{H_3(\lambda_l)}, g_2).$$

If the result is true, the TPA can make sure the user's data is correct in CSS. Otherwise, the shared data is incorrect.

Dynamic data phase. In the phase, we will only describe inserted operation because it is more difficult than update and deletion. Then, we also describe their defined form of index hash tables (IHT) (as shown in Figure 2). The IHT has four columns which are described by indexes $id_j = \{v_j, r_j\}$, blocks m_j, virtual index $v_j = j\delta$ (where $\delta \in N^*$ is a system parameter by the original user) and random value $r_j = H_2(m_j || v_j)$, for $j \in [1, n]$ is the number of block.

Step 1: The user wants to insert a new block m'_j into shared data. The user computes the new identifier of the block $id'_j = \{v'_j, r'_j\}$, where $v'_j = \lfloor (v_{j-1} + v_j)/2 \rfloor$ and $r'_j = H_2(m'_j || v'_j)$. The user computes $\beta'_j = H_1(id'_j) \prod_{l=1}^{k} \eta_l^{m_{j,l}}$, computes $\sigma'_{j,s} = (\frac{\beta'_j}{\Psi(\prod_{i \neq s} w_i^{a_{j,i}})})^{1/x^s}$ and generates a new ring signature. Finally, the user uploads $\{m'_j, id'_j, v'_j, r'_j, \sigma'_{j,s}\}$ to the CSS.

Step 2: The CSS updates index hash table (IHT) where the new block m'_j is inserted in the virtual index v'_j and the total number of blocks increase $n + 1$ in shared blocks (as shown in Figure 3).

User revocation phase. Because Wang et al.'s originally intended to solve the privacy-preserving public auditing mechanism for shared data, their scheme needs to decide in advance the number of group members and computes the number of keys. Therefore, their scheme is a static group model which does not consider the situation of a new user to be added in the group or an existing user to be revoked from the group. In order to support a dynamic group, they propose an improved solution where the ring signature on shared data need to re-compute the signer's private key and all the current users' public key when the membership of the group is changed.

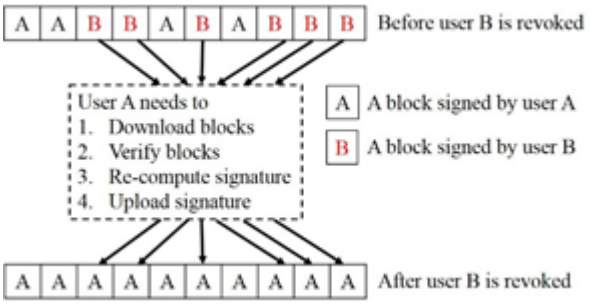

Figure 4: The traditional approach of user A and user B share data in the cloud

For instance, the number of group member is d. When a new user u_{d+1} is added into the group, the signer needs to re-compute his/her private key and others need to re-compute their public keys $(pk_1, \cdots, pk_{d+1})$ on the ring signature. When an existing user u_d is revoked from the group, the signer needs to re-compute his/her private key and others need to re-compute their public keys $(pk_1, \cdots, pk_{d-1})$ on the ring signature.

However, to satisfy the requirement of dynamic group, users need to pay a large amount of computation in the re-computation (as Setup phase). If the group has a lot of users and the user are frequently added or revoked in the group, users need to re-execute the setup phase. Therefore, how to effectively solve the re-computation of dynamic group will be a serious issue.

3.2 Wang et al.'s Scheme

Wang et al. [11] proposed a novel public auditing for shared data with efficient user revocation in the cloud (as Panda). They consider a situation that a user is revoked in the group because of the user's malicious behavior. However, the user is revoked before his/her signature blocks cannot find corresponding blocks of the signer. Therefore, these signed blocks of signature needed to be re-signing. The traditional approach explained that these blocks of the revoked user B gives existing user A to download, verify, re-sign and upload the re-signed blocks (as shown in Figure 4). However, it is not an efficient approach which will increase the exiting users' burden on communication and computation of resources.

Therefore, they utilized the concept of proxy re-signature [5] to solve public auditing for shared data with user revocation. This is not needed to spend a lot of resources on the existing users because the existing users can delegate a cloud server to re-sign blocks by generating re-signed key on the proxy re-signature model (as shown in Figure 5). Their scheme is efficient on user revocation

Index	Block	Virtual index	Random value
1	m_1	1δ	r_1
2	m_2	2δ	r_2
3	m_3	3δ	r_3
...	...	...	...
n	m_n	$n\delta$	r_n

insert

Index	Block	Virtual index	Random value
1	m_1	1δ	r_1
2	m_2'	$\lfloor 3\delta/2 \rfloor$	r_2'
3	m_2	2δ	r_2
4	m_3	3δ	r_3
...	...	...	...
$n+1$	m_n	$n\delta$	r_n

Figure 3: Insert block m_2' into shared data by using the index hash table (IHT) as identifiers on Oruta

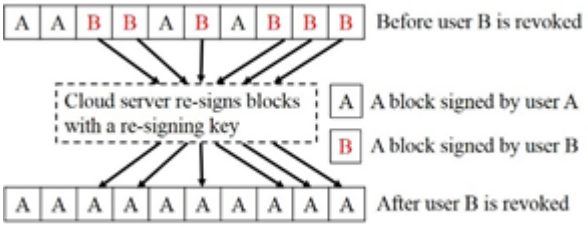

Figure 5: Wang et al.'s approach of user A and user B sharing data in the cloud

because it can reduce the computation and communication resources of existing users.

However, this solution extends an important issue for the semi-trusted cloud server to manage the re-signing key of the group. In order to avoid the single re-signing proxy on the semi-trusted cloud server, they proposed a solution which utilized a new multi-proxy model by improved Shamir Secret Sharing proxy model [44]. Because this multi-proxy model is not included in this paper, this issues more detail reference [11]. They utilized index hash tables (IHT) from Zhu et al.'s scheme [56] to support dynamic data. Finally, they extend their mechanism to support batch auditing which can allow public auditing on different users simultaneously and improve the efficiency of verification for multiple auditing tasks.

Next we will describe their scheme including setup, public auditing, dynamic data and user revocation phase. Before executing each phase, the CSS need to generate the global parameters: $(e, p, q, G_1, G_T, g_1, w, H_1, H_2, d, n)$. Their scheme is as follows:

Setup phase. In the phase, we will describe key generation and signature.

Step 1: A user u_i randomly chooses $x_i \in Z_p$ and computes the user's public key $pk_i = g^{x_i}$, and private key is $sk_i = x_i$. If the user is original, he/she will create a user list which includes the identity of all users in the group and the user list is public and signed by the original user.

Step 2: The user u_i uses private key $sk_i = x_i$ to sign the shared data block $m_j \in Z_p$ and its block identifier id_j, and w be another generator of G_1, where $j \in [1, n]$. Finally, the signature block is $\sigma_j = (H_1(id_j)w^{m_j})^{x_i} \in G_1$.

Public auditing phase. In the phase, we will describe challenge request, proof generation and proof verification.

Step 1: The TPA selects c elements as a subset L of set $[1, n]$ chooses a random value $y_l \in Z_q$, for $l \in L$ and q is a much smaller prime than p. Then, the TPA sends the challenged message $\{(l, y_l)_{l \in L}$ to the CSS.

Step 2: The CSS divides set L into d subset $(L_1, \cdots, L_d)$, where L_i is the subset of selected blocks signed by user u_i. Then the CSS computes $\mu_i = \sum_{l \in L_i} y_l m_l \in Z_p$. Finally, the CSS computes $\phi_i = \prod_{l \in L_i} \sigma_l^{y_l} \in G_1$, for $i \in [1, d]$ before the CSS returns an auditing proof $\{\{\mu_1, \cdots, \mu_d\}, \{\phi_1, \cdots, \phi_d\}, \{id_l, s_l\}_{l \in L}\}$.

Step 3: The TPA uses an auditing challenge $\{(l, y_l)\}_{l \in L}$, an auditing proof $\{\{\mu_1, \cdots, \mu_d\}, \{\phi_1, \cdots, \phi_d\}, \{id_l, s_l\}_{l \in L}\}$ and all the group members' public keys $(pk_1, \cdots, pk_d)$ to check the correctness of the auditing proof by computing the equation $e(\prod_{i=1}^{d} \phi_i, g) \stackrel{?}{=} \prod_{i=1}^{d} e(\prod_{l \in L_i} H_1(id_l)^{y_l} w^{\mu_i}, pk_i)$. If the result is true, the TPA can make sure the user's data is correct in CSS. Otherwise, the shared data is incorrect.

Dynamic data phase. In the phase, we will only describe inserted operation because it is more difficult than update and deletion. Then, we also describe their defined form of index hash tables (IHT) (as shown in Figure 6). The IHT has four columns which are described by indexes $id_j = \{v_j || r_j || s_j\}$, blocks m_j, virtual index $v_j = \text{j} \cdot \delta$ (where $\delta \in N^*$ is a system parameter by the original user), random value $r_j = H_2(m_j || v_j)$ and s_j is the signer identity of block m_j, for $j \in [1, n]$ is the number of block.

Step 1: The user s_i' wants to insert a new block m_j' into shared data. The user s_i' computes the new identifier of the block $id_j' = \{v_j' || r_j' || s_i'\}$, where

Index	Block	Virtual index	Random value	Signer id
$id_j = \{v_j \| r_j \| s_j\}$	m_j	$v_j = j \cdot \delta$	$r_j = H_2(m_j \| v_j)$	s_j

Figure 6: Wang et al.'s index hash table (IHT) on Panda's mechanism

$v'_j = \lfloor (v_{j-1} + v_j)/2 \rfloor$ and $r'_j = H_2(m'_j \| v'_j)$. The user uses his/her private key $sk_i = x_i$ to sign the block $\sigma'_j = (H_1(id_j)w^{m_j})^{x_i} \in G_1$. Finally, the user uploads $\{m'_j, id'_j, v'_j, r'_j, \sigma'_j, s_i\}$ to the CSS.

Step 2: The CSS updates index hash table (IHT) where uses $\{id'_j, m'_j, v'_j, r'_j, s'_i\}$ instead of $\{id_j, m_j, v_j, r_j, s_i\}$, stores the signature σ'_j instead of σ_j and the total number of blocks increases $n+1$ in shared blocks (as shown in Figure 7).

User revocation phase. In the phase, we will describe rekey generation and re-signature. For example, the user u_i is revoked with the signature of the user u_j instead of the signature of the user u_i.

Step 1: The CSS chooses a random value $r \in Z_p$ and sends to the user u_i.

Step 2: The user u_i computes r/x_i and sends to the user u_j.

Step 3: The user u_j computes $(rx_j)/x_i$ and sends to the CSS.

Step 4: The CSS generates a re-signing key $?rk_{i\to j} = x_j/x_i \in Z_p^*$. The CSS use the user u_i's public key, the signature σ_k, the block m_k and the block identifier id_k to compute $e(\sigma_k, g) \stackrel{?}{=} e(H_1(id_k)w^{m_k}, pk_i)$ and check the signature σ_k whether the block m_k was signed by the user u_i. If the result is true, the CSS computes $\sigma'_k = \sigma_k^{rk_{i\to j}} = (H_1(id_k)w^{m_k})^{x_i x_j/x_i} = (H_1(id_k)w^{m_k})^{x_j}$, otherwise the CSS aborts the re-signed request.

Step 5: The original user updates the user u_j's id instead of the user u_i's id on the singer identifier from a user list and signs the new user list.

3.3 Yuan and Yu's Scheme

Yuan and Yu [59] proposed a novel public integrity auditing for dynamic data sharing scheme which supports multiple users to modify shared data in the cloud storage service. They considered a problem where the cloud server aggregates authenticated tags from multiple users in public auditing phase. Because the data blocks can be modified and signed by different users' secret keys which are different each other, the cloud server has to one by one verify different users' signature in public auditing phase.

For example, a simple method can solve the problem where all users of the group share the same secret key, so it can be easily aggregated. However, when a user is revoked, he/she still can generate authenticated tags. Therefore, they utilized polynomial commitment scheme [58] to design a polynomial-based authentication tags from multiple users into one which can send the integrity proof information to the public verifier. Therefore, the public verifier only needs a constant size of integrity proof information and a constant number of computational operations. Finally, they extend their mechanism to support batch auditing which can allow public auditing on different users simultaneously and improve the efficiency of verification for multiple auditing tasks. Next we will describe their scheme including setup, public auditing, dynamic data and user revocation phase. Before executing each phase, the CSS need to generate the global parameters: $(e, p, q, G_1, G_T, g_1, u, H_2, d, n)$. Their scheme is as follows:

Setup phase. In the phase, we will describe key generation and signature.

Step 1: The master user u_0 responsibly manages the membership of the group and generates public keys (PK), users' secret keys (SK) and the system's master key (MK). The master user randomly chooses $\{x_i\}_{1\le i\le d-1} \in Z_q^*$, $\alpha \in Z_q^*$ and computes $v = g^{\alpha x_0}$, $k_0 = g^{x_0}$, $\{k_i = g^{x_i}, g^{x_0/x_i}\}_{1\le i\le d-1}$. The public keys are $PK = \{g, u, q, v, \{g^{\alpha^j}\}_{0\le j\le k+1}, k_0, \{k_i, g^{x_0/x_i}\}_{1\le i\le d-1}\}$, the master key is $MK = \{x_0, \alpha\}$ and the secret keys are $SK_i = \{x_i\}_{1\le i\le d-1}$.

Step 2: The master user u_0 computes the signature block $\sigma_i = (u^{B_i} \prod_{j=0}^{k} g^{m_{ij}\alpha^{j+2}})^{x_0} = (u^{B_i} g^{f_{\rightharpoonup \beta_i}(\alpha)})^{x_0}$ where $\rightharpoonup \beta_i = \{0, 0, \beta_{i,0}, \beta_{i,1}, \cdots, \beta_{i,k-1}\}$ and $\beta_{i,j} = m_{i,j}$. Then, $B_i = H_2(\{fname \| i \| t_i \| d\})$, $fname$ is the file name, i is the index of data block m_i, t_i is the time stamp and d is the index of user in the group. Finally, the master user sends $\{m_i, \sigma_i]\}_{1\le i\le n}$ to the CSS and sends $\{B_i\}_{1\le i\le n}$ to the TPA.

Public auditing phase. In the phase, we will describe challenge request, proof generation and proof verification.

Step 1: The TPA selects c data blocks as a subset L and chooses two random values $R \in Z_q^*$ and $\mu \in Z_q^*$. Then, the TPA computes $X = \{(g^{x_0/x_i})^R\}_{0\le i\le d-1}$ and g^R. The TPA sends the challenge message $CM = \{L, X, g^R, \mu\}$ to the CSS.

Step 2: The CSS generates $\{p_i = \mu^i \bmod q\}_{i\in L}$ and computes $y = f_{\rightharpoonup A}(\mu) \bmod q$, where $\rightharpoonup A = \{0, 0, \sum_{i\in L} p_i m_{i,0}, \cdots, \sum_{i\in L} p_i m_{i,k-1}\}$. The CSS divides the polynomial $f_{\rightharpoonup A}(x) - f_{\rightharpoonup A}(\mu)$ with $(x-\mu)$ using polynomial long division, and indicates the coefficients vector of the resulting quotient polynomial as

Index	Block	Virtual index	Random value	Signer id
1	m_1	1δ	r_1	s_1
2	m_2	2δ	r_2	s_2
3	m_3	3δ	r_3	s_3
...	...	...	...	...
n	m_n	$n\delta$	r_n	s_n

insert

Index	Block	Virtual index	Random value	Signer id
1	m_1	1δ	r_1	s_1
2	m_2'	$\lfloor 3\delta/2 \rfloor$	r_2'	s_2'
3	m_2	2δ	r_2	s_2
4	m_3	3δ	r_3	s_3
...	...	...	...	...
$n+1$	m_n	$n\delta$	r_n	s_n

Figure 7: Insert block m_2' into shared data by using the index hash table (IHT) as identifiers on Panda

$\rightharpoonup w = (w_0, w_1, \cdots, w_k)$, that is $f_{\rightharpoonup w} \equiv \frac{f_{\rightharpoonup A}(x) - f_{\rightharpoonup A}(\mu))}{(x-\mu)}$. The CSS computes $\Psi = \prod_{(} j = 0)^k (g^{\alpha^j})^{w_j} = g^{f_{\rightharpoonup w}(\alpha)}$. Then, the data blocks of a challenged subset L are modified by the user $\{u_s\}_{s \in d}$, the CSS computes $\pi_i = (\sigma_i, g^{x_0 R/x_s}) = e((u^{B_i} g^{f_{\rightharpoonup \beta_i}(\alpha)}), g)^{x_0 R}$ or modified by the master user u_0, the CSS computes $\pi_i = e(\sigma_i, g^R) = e((u^{B_i} g^{f_{\rightharpoonup \beta_i}(\alpha)}), g)^{x_0 R}$. These π_i will be aggregated as $\pi = \prod_{i \in L} \pi_i^{p_i}$. Finally, the CSS returns the proof message $\{\pi, \Psi, y\}$ to the TPA.

Step 3: The TPA computes $\eta = u^{\omega}$, where $\omega = \sum_{i \in L} B_i p_i$ and verifies the integrity of file as $e(\eta, k_0^R) e(\Psi^R, v \cdot k_0^{-\mu}) \stackrel{?}{=} \pi \cdot e(k_0^{-y}, g^R)$. If the result is true, the TPA can make sure the user's data is correct in CSS. Otherwise, the shared data is incorrect.

Dynamic data phase. In the phase, we will only describe update operation because they have not considered fully dynamic operation such as the data block insert and delete operation.

Step 1: A user u_s of group wants to modify a block m_i to m_i'. Therefore, u_s needs to use own secret key x_s to compute the re-signed block $\sigma_i' = (u^{B_i'} \prod_{j=0}^{k-1} g^{m_{i,j}' \alpha^{j+2}})^{x_s} = (u^{B_i'} g^{f_{\rightharpoonup \beta_i'}(\alpha)})^{x_s}$, where $\rightharpoonup \beta_i' = \{0, 0, \beta_{i,0}', \beta_{i,1}', \cdots, \beta_{i,k-1}'\}$ and $\beta_{i,j}' = m_{i,j}'$. Then $B_i' = H(fname||i||t_i'||d)$. Finally the user u_k uploads $\{m_i', \sigma_i'\}$ to the CSS and uploads B_i' to the TPA.

Step 2: The CSS receives the modified message and uses $\{m_i', \sigma_i'\}$ instead of $\{m_i, \sigma_i\}$.

Step 3: The TPA receives the modified message and uses B_i' instead of B_i.

User revocation phase. In the phase, we will describe rekey generation, reject generation and re-signature.

Step 1: When a user u_s of group is revoked, the master user u_0 computes rekey generation $\chi = \frac{x_0 + \rho}{x_s} \bmod q$, where $\rho \in Z_q^*$ is a random value and computes reject generation $g^{\frac{x_0}{(x_0+\rho)}}$. Finally, the master user u_0 sends χ to the CSS and sends $g^{\frac{x_0}{(x_0+\rho)}}$ to the TPA and group users.

Step 2: The CSS receives χ and updates the signature $\sigma_i' = \sigma_i^{\chi} = (u^{B_i} g^{f_{\rightharpoonup \beta_i}(\alpha)})^{x_0 + \rho}$

Step 3: The TPA and group users reject the user u_s's public parameter g^{x_0/x_s}.

3.4 Jiang et al.'s Scheme

Jiang et al. [28] proposed a public integrity auditing for shared dynamic cloud data with group user revocation. They considered a problem of collusion attack where a revoked user can collude with the malicious cloud server to change the group existed user's data. Because a group user may have malicious behavior, the data owner (or the group manager) will revoke the group of malicious user. However, if a semi-trusted cloud server cooperates with the revoked user each other, the group users' data will have a secure problem.

Therefore, how to design an efficient and reliable scheme, while achieving secure group user revocation. They propose a mechanism which not only supports the group data encryption and decryption during the data modification processing, but also achieves efficient and secure user revocation. They utilized a vector commitment scheme [17], utilized an asymmetric group key agreement (AGKA) scheme [55] and a verifier-local revocation group signature scheme [7] to construct their mechanism. A vector commitment scheme is used over the database, and AGKA scheme is used to encrypt/decrypt the share database, and a verifier-local revocation group signature scheme will avoid the collusion of cloud and revoked group users.

Next we will describe their scheme including setup, public auditing, dynamic data and user revocation phase. Before executing each phase, the CSS need to generate the public parameters: $PP = (p, q, G, G_T, H, g, \{h_i\}_{i \in n}, \{h_{i,j}\}_{i,j \in n, i \neq j})$. For all $i = (1, 2, \cdots, n)$, set $h_i = g^{z_i}$. For all i,j= $(1, 2, \cdots, n)$, set $h_{i,j} = g^{z_i z_j}$ where random values $z_i = (z_1, z_2, \cdots, z_n) \in Z_p$. Their scheme is as follows:

Setup phase. In the phase, we will describe key generation, commitment and signature.

Step 1:

1) The data owner chooses a random value $\gamma \in Z_p^*$, computes $w = g_2^{\gamma}$ and generates the

group public key $gpk = (g_1, g_2, w)$. Then the value γ is only known and protected by the data owner.

2) The data owner generates an SDH (Strong Diffie-Hellman) tuple (A_i, x_i) by choosing random values $x_i \in Z_p^*$ for each user, such that $\gamma + x_i \neq 0$ and computing $A_i = g_1^{1/(\gamma+x_i)}$. Then, the users of group generate the group secret key $gsk = (gsk[1], gsk[2], \cdots, gsk[d])$ where $gsk[i] = (A_i, x_i)$, and the revocation token RL corresponding to a user's secret key is $grt[i] = A_i$.

3) Finally, the user of group create (gpk, gsk, grt).

Step 2: The data owner computes commitment $C = (h_1^{m_1} \cdot h_2^{m_2} \cdots h_n^{m_n}) = \prod_{i=1}^{n} h_i^{m_i}$ and auxiliary information $aux = (m_1, m_2, \cdots, m_n)$.

Step 3:

1) For t^{th} time, the data owner updates data after the data blocks is signed. The data owner chooses a random value $r \in Z_p$ and obtain generators $(\hat{u}, \hat{v})$ in G_2 from H_0 as $(\hat{u}, \hat{v}) = H_0(gpk, \{C(t-1), C^t, t\}, r) \in G_2^2$ and computes $(\hat{u}, \hat{v})$ images in G_1 as $u = \Psi(\hat{u})$, $v = \Psi(\hat{v})$.

2) The data owner chooses an exponent $\alpha \in Z_p$ and compute $T_1 = u^{\alpha}$ and $T_2 = A_i v^{\alpha}$, sets $\delta = x_i \alpha \in Z_p$ and pick blinding values r_α, r_x and $r_\delta \in Z_p$, computes helper values $R_1 = u^{r_\alpha}$, $R_2 = e(T_2, g_2)^{r_x} \cdot e(v, w)^{-r_\alpha} \cdot e(v, g_2)^{-r_\delta}$ and $R_3 = T_1^{r_x} u^{-r_\delta}$, computes a challenge value c from H_2 as $c = H_2(gpk, (C(t-1), C^t, t), r, T_1, T_2, R_1, R_2, R_3) \in Z_p$, computes $s_\alpha = r_\alpha + c\delta$, $s_x = r_x + cx_i$ and $s_\delta = r_\delta + c\delta \in Z_p$. Finally, the data owner sends the signature $\sigma^t = (r, T_1, T_2, c, s_\alpha, s_x, s_\delta)$ to the CSS.

3) If σ^t is vaild, then the CSS computes $C(t) = \sigma^t C^t$ and adds the information of $\sum(t) = (C(t-1), C^t, t, \sigma^t)$ to aux.

4) Set public key parameter $PK = (PP, gpk, C(t-1), C(t))$.

Public auditing phase. In the phase, we will describe TPA to verify the validity of the signature.

Step 1: The CSS sends $(gpk, \sigma^t, C(t-1), C^t, t)$ to the TPA.

Step 2: Because the signature is $\sigma^t = (r, T_1, T_2, c, s_\alpha, s_x, s_\delta)$ and group public key is $gpk = (g_1, g_2, w)$, the TPA can computes $\hat{u}$, $\hat{v}$ and their image $u = \Psi(\hat{u}$, $v = \Psi(\hat{v}$ where $(\hat{u}, \hat{v}) = H_0(gpk, \{C(t-1), C^t, t\}, r) \in G_2^2$ and compute helper values $R_1' = u^{s_\alpha}/(T_1^c)$, $R_2' = e(T_2, g_2)^{s_x} \cdot e(v, w)^{-s_\alpha} \cdot e(v, g_2)^{-s_\delta} \cdot (e(T_2 \cdot w)/e(g_1/g_2))^c$ and $R_3' = T_1^{s_x} \cdot u^{-s_\delta}$. Then, the TPA computes a challenge value $c' \in Z_p$ using H_2 as $c' = H_2(gpk, (C(t-1), C^t, t), r, T_1, T_2, R_1', R_2', R_3')$ and checks the challenge $c \stackrel{?}{=} c'$.

Dynamic data phase. In the phase, we will describe update operation because they have not considered fully dynamic operation such as the data block insert and delete operation.

Step 1: Jiang et al. assumed that the current public key is $PK = (PP, gpk, C(t-1), C(t))$. A group user uses the public key PK to compute a proof $\Lambda_i^t = \prod_{j=1, j\neq i}^{n} h_{i,j}^{m_j^t} = (\prod_{j=1, j\neq i}^{n} h_j^{m_j^t})^{z_i}$ of the i^{th} committed message and sends $\tau = (m_i^t, \Lambda_i^t, \sum(t))$ to the CSS.

Step 2: The CSS will verify whether the user is revoked in the group.

1) The CSS receives the i^{th} committed message $\tau = (m_i^t, \Lambda_i^t, \sum(t))$. Then, the CSS sends $(gpk, \sigma^t, \sum(t))$ to the TPA.

2) Because the signature is $\sigma^t = (r, T_1, T_2, c, s_\alpha, s_x, s_\delta)$ and group public key is $gpk = (g_1, g_2, w)$, the TPA can computes $\hat{u}$, $\hat{v}$ and their image $u = \Psi(\hat{u})$, $v = \Psi(\hat{v})$ where $(\hat{u}, \hat{v}) = H_0(gpk, \{C(t-1), C^t, t\}, r) \in G_2^2$ and compute helper values $R_1' = u^{s_\alpha}/(T_1^c)$, $R_2' = e(T_2, g_2)^{s_x} \cdot e(v, w)^{-s_\alpha} \cdot e(v, g_2)^{-s_\delta} \cdot (e(T_2 \cdot w)/e(g_1/g_2))^c$ and $R_3' = T_1^{s_x} \cdot u^{-s_\delta}$. Then, the TPA computes a challenge value $c' \in Z_p$ using H_2 as $c' = H_2(gpk, (C(t-1), C^t, t), r, T_1, T_2, R_1', R_2', R_3')$ and checks the challenge $c \stackrel{?}{=} c'$. The TPA ensures that σ^t was not generated by each revoked user $A \in RL$.

3) If the result is true, the CSS need to verify the correctness of the group user. The CSS checks the equation $e(C^t/(h_i^{m_i^t}), h_i) \stackrel{?}{=} e(\Lambda_i^t, g)$. If the result is true, which means Λ_i^t is a valid proof that C^t was created to a sequence $(m_1, m_2, \cdots, m_n)$, such that $m = m_i$.

Step 3: A group user wants to update message m_i' instead of m_i, computes the updated commitment $C' = C \cdot h_i^{m'-m}$ and the updated information $U = (m, m', i)$.

Step 4: The TPA can computes an update proof $\Lambda_j = \prod_{i=1, i\neq j}^{n} h_{i,j}^{m_i} = (\prod_{i=1, i\neq j}^{n} h_i^{m_i})^{z_j}$ and j is the position of message. The proof Λ_j is valid with respect to C' which contains m' as the new message at position j. The TPA uses the update information $U = (m, m', i)$ to generate the proof of update. If the position of message $i \neq j$, compute the updated commitment $C' = C \cdot h_i^{m'-m}$ and the updated proof is

$\Lambda'_j = \Lambda_j \cdot (h_i^{m'-m})^{z_j} = \Lambda_j \cdot h_{j,i}^{m'-m}$. If the position of message $i = j$, compute the updated commitment $C' = C \cdot h_i^{m'-m}$ while not changing the proof Λ_i. Finally, the TPA verifies the commitment C' and corresponding proof Λ_i is also valid over message m'_i.

User revocation phase. In the phase, we will describe TPA to verify the validity of the signature and check the revocation list.

Step 1: The CSS sends $(gpk, \sigma^t, C(t-1), C^t, t)$ to the TPA.

Step 2: Because the signature is $\sigma^t = (r, T_1, T_2, c, s_\alpha, s_x, s_\delta)$ and group public key is $gpk = (g_1, g_2, w)$, the TPA can computes $\hat{u}$, $\hat{v}$ and their image $u = \Psi(\hat{u})$, $v = \Psi(\hat{v})$ where $(\hat{u}, \hat{v}) = H_0(gpk, \{C(t-1), C^t, t\}, r) \in G_2^2$ and compute helper values $R'_1 = u^{s_\alpha}/T_1^c)$, $R'_2 = e(T_2, g_2)^{s_x} \cdot e(v, w)^{-s_\alpha} \cdot e(v, g_2)^{-s_\delta} \cdot (e(T_2 \cdot w)/e(g_1/g_2))^c$ and $R'_3 = T_1^{s_x} \cdot u^{-s_\delta}$. Then, the TPA computes a challenge value $c' \in Z_p$ using H_2 as $c' = H_2(gpk, (C(t-1), C^t, t), r, T_1, T_2, R'_1, R'_2, R'_3)$ and checks the challenge $c \stackrel{?}{=} c'$.

Step 3: The TPA ensures that σ^t was not generated by each revoked user $A \in RL$. Therefoere, the TPA checks whether A is encoded in (T_1, T_2) by checking if $e(T_2/A, \hat{u}) \stackrel{?}{=} e(T_1, \hat{v})$. If no element of RL is written in (T_1, T_2), the signer of σ^t has not been revoked.

4 Analysis

In the section, we will analyze these schemes [8, 11, 28, 59] which contain functional requirement, security and performance. And we also use the tables to present a corresponding requirement in each scheme.

4.1 Functional Evaluation

In Table 2, we will analyze seven functional requirements: blockless verification, stateless verification, bath auditing, dynamic data, anonymity, privacy presenting and user revocation in the representative approaches. Yuan, Yu's scheme [28] and Jiang et al.'s scheme [59] decided which TPA needed to maintain the data situation of the user in the dynamic data phase. Because Jiang et al.'s scheme has not consider blockless verification, their scheme extends to support batch auditing which is difficult in the public auditing phase. In the dynamic data phase, Yuan, Yu's scheme and Jiang et al.'s scheme only considered data update, so their scheme can support data insert operation. Because Wang et al.' scheme [8] utilized ring signature, their scheme can influence the TPA to get the user identity. These scheme can satisfy privacy presenting, when the TPA can get the data of the group in the public auditing phase. Because Wang et al. [8] first proposed the scheme which can support shared data on public auditing, they have not considered to user revocation in the group. However, they make up the problem, but the improved scheme needs to re-generate the key of each user. These approach can satisfy the requirement of user revocation.

4.2 Security Evaluation

In Table 3, we will analyze the five attack models: inside attack, forge attack, replace attack, impersonation attack and collusion attack in the representative approaches. In the inside attack, because these schemes are used by the user to upload plaintext in the cloud storage server, the cloud storage server can know the user's data. Therefore, the cloud storage server can use unauthorized data. In the forge attack, these scheme can resist the cloud server to forge the data tag of data block because the data owner upload the signed the data tag of data block. Therefore, the cloud server is hard to forge a legitimate data tag. In the replace attack, Wang et al. [8] and Wang et al. [11] do not consider the cloud server does not update the user's data, so they cannot support the replace attack. Yuan et al.'s scheme uses the time stamp to record updated time, so it can check the time stamp of the data. Jiang et al.'s scheme considered which TPA verifies the update proof, so it can check whether the cloud server update the user's data. In the impersonation attack, because they focused on data integrity, their scheme do not consider authentication. Jiang et al.'s scheme was only a simple authentication where the TPA verified the user on the revocation list. However, when the user does not exist on the revocation list, the TPA cannot verify the impersonation attack. In the collusion attack, because Wang et al.'s scheme [8] has not supported user revocation, the revoked user and the semi-trusted server can collude to attack the shared data. Because Wang et al. [11], Yuan et al. and Jiang et al. have support user revocation, they can avoid a collusion attack.

4.3 Performance Evaluation

We will analyze four phases: setup phase, public auditing phase, dynamic data phase and user revocation phase in the four entities which include data owner, user (the group user), cloud storage server (CSS) and third party auditor (TPA). Before we analyze the performance evaluation, first we introduce the notations in Table 4.

In Table 5, we analyze four schemes how to execute a setup phase. Wang et al.'s scheme [11] explained that the data owner needs lower computing resource in the setup phase. The group user does not need to generate a secret key because the data owner supports key generation as shown in the Jiang et al.'s scheme.

In Table 6, we analyze four scheme how to execute a public auditing phase. Because in a public auditing phase the data owner and the group user have not to execute,

Table 2: Comparison of functional requirements

	Wang et al. [8]	Wang et al. [11]	Yuan, Yu [59]	Jiang et al. [28]
Blockless verification	Yes	Yes	Yes	No
Stateless verification	Yes	Yes	No	No
Batch auditing	Yes	Yes	Yes	No
Dynamic data	Yes	Yes	Partial	Partial
Anonymity	Yes	No	No	No
Privacy Presenting	Yes	Yes	Yes	Yes
User Revocation	No	Yes	Yes	Yes

Table 3: Comparison of security attack

	Wang et al. [8]	Wang et al. [11]	Yuan, Yu [59]	Jiang et al. [28]
Inside attack	No	No	No	No
Forge attack	Yes	Yes	Yes	Yes
Replace attack	No	No	Yes	Yes
Impersonation attack	No	No	No	No
Collusion attack	No	Yes	Yes	Yes

Table 4: Notations

Notation	Significance
T_E/T_D	The computing time of asymmetric encryptions
T_{Ge}	The computing time of exponentiation in group operation
T_{BLS}	The computing time of BLS signature
T_B	The computing time of bilinear pairing
T_M	The computing time of multiplication
T_A	The computing time of addition
T_{GM}	The computing time of multiplication in group operation
T_h	The computing time of hash function
n	The number of block in a file
i	The number of verified block
d	The total number of users in the group
k	A block element of the shared data block m_i

Table 5: Comparison of computation in setup phase

	Wang et al. [8]	Wang et al. [11]	Yuan, Yu [59]	Jiang et al. [28]
Data owner	$n(T_{Ge}^{d-1}+T_{Ge}^{k}+T_{BLS}$ $+2T_{GM}+T_h)+T_{Ge}$ $+2T_{GM}+T_h)+T_{Ge}$	$(n+1)T_{Ge}$ $+n(T_{BLS}+T_{GM}+T_h)$ $+n(T_{BLS}+T_{GM}+T_h)$	$n(T_{Ge}^{k}+T_{Ge}+T_{BLS}$ $+T_{GM}+T_h)$ $+2T_{Ge}+T_M$	$(d+6)T_{Ge}+T_{Ge}^{n}$ $+(d+3)T_A+4T_M$ $+4T_{GM}+2T_h$
User	T_{Ge}	T_{Ge}	$2T_{Ge}$	-
CSS	-	-	-	T_{GM}
TPA	-	-	-	-

they do not consume any computing resources. In the Jiang et al.'s scheme, the CSS does not consume computing resource because the CSS will response the entire data to the TPA. Then, the TPA needs to verify the entire data, so the TPA requires more computing resources. The CSS is an affected factor including a block element of the shared data block, the total number of users in the group and the number of verified block. Wang et al. [8] considered two factors include the number of verified block and a block element of the shared data block, Wang et al. [11] considered one factor which is the total number of users in the group and Yuan et al. considered all factors. However, Yuan et al.'s scheme is more flexible in different situations.

In Table 7, we analyze four schemes how to execute a dynamic data phase. Jiang et al.'s scheme obviously requires more computing resources because their scheme has three entities needed to execute. These schemes [8, 11, 59] only consider that the group user transfers the updated data to the CSS, and the CSS directly update the data. Therefore, when the semi-trusted CSS has not updated the data, the group user cannot get related message. Jiang et al.'s scheme considered that the CSS can verify whether the user is in this group and the TPA can verify whether the stored data of the CSS has been updated. Therefore, Jiang et al.'s scheme spent a lot of computing resources in the verification.

In Table 8, we analyze four schemes how to execute a user revocation phase. Wang et al.'s scheme [11] and Yuan et al.'s scheme are similar because they scheme which data owner delegates the CSS to re-sign the signed data block of the revoked user. However, Yuan et al.'s scheme decided which data owner needs more computing resources. Jiang et al.'s scheme needed to verify the signature and check whether the revoked user has been revoked in the revocation list RL.

In Table 9, we analyze four schemes how to distribute storage. The TPA cannot require storage space in Wang et al.'s scheme [8] and Wang et al.'s scheme [11]. Yuan et al.'s scheme decided which TPA requires to store file information, and Jiang et al.'s scheme decided which TPA requires to store revocation list. Because Wang et al.'s scheme [8] considered anonymity, the CSS could not store the signer's information. However, Wang et al.'s scheme [11] considered user revocation, so the CSS requires to update the signed data block of the revoked user. Because Yuan et al.'s scheme only consider modification operation, the CSS only store m and σ. However, Jiang et al.'s scheme is $aux = (m_1, m_2, \cdots, m_n)$, $(C(t-1), C^t, t, \sigma^t)$, so their scheme requires more storage space.

5 Conclusion and Future Work

In the cloud storage service, the data integrity of remote verification is already a critical issue. The concept of public audit can solve to remotely verify data integrity and extend to verify the shared data. We organize public auditing requirements containing function, security and performance from the many relevant literatures. We also list the four representative approaches and analyze these approaches. These comparison tables can clearly understand the advantages and disadvantages of each approach. Finally, in this paper, we provide the future development of public audit and shared data.

For future developments, we will focus on the following areas of particular interest. Efficiency: because users demand high performance, the scheme satisfies an effective scheme to reduce the computing resources which include public audit, dynamic data and user revocation of the operation. Therefore, how to design an efficient public audits with shared data that is an important issue.

Security: in addition to data integrity, the public auditing need to consider the data confidentiality. Because the user will store data in the cloud storage service, the cloud service provider can access the user's data. Therefore, the user need to encrypt data before the user uploads data to the cloud storage service. How to design a public audit with shared data in the situation of encrypted data which will be able to satisfy integrity and confidentiality simultaneously.

Data recovery: the user upload data to the cloud storage service before the user deletes data which will reduce the user's storage space. When the cloud server is lost the user data, third party auditor verifies the user's data is in complete. However, the user do not back up data on the local storage space. Therefore, the user need to save his/her data. How to design a scheme which can support public audit and data recovery.

References

[1] M. Armbrust, et al., "A view of cloud computing", *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 598–609, Virginia, USA, 2007.

[3] G. Ateniese, et al., "Remote data checking using provable data possession", *ACM Transactions on Information and System Security*, vol. 14, no. 1, pp. 1–34, 2011.

[4] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks*, pp. 9:1–9:10, Istanbul, Turkey, 2008.

[5] M. Blaze, G. Bleumer, M. Strauss, "Divertible protocols and atomic proxy cryptography", in *Advances in Cryptology (EUROCRYPT'98)*, LNCS 1403, pp. 127–144, Springer, 2006.

[6] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proceedings of the 7th*

Table 6: Comparison of computation in Public auditing phase

	Wang et al. [8]	Wang et al. [11]	Yuan, Yu [59]	Jiang et al. [28]
CSS	$1T_{Ge}$ $+k((i+1)T_M+T_A+T_h)$	$(T^d_{Ge}+dT_M$ $(T^d_{Ge}+dT_M$	$iT_{Ge}+2T^k_{Ge}+T^i_{Ge}$ $+k(iT_M)+T_B$	-
TPA	T_B	T_B	$(d+1)T_{Ge}+iT_M+T_B$	$4T_{Ge}+2T_h+5T_{GM}$

Table 7: Comparison of computation in Dynamic data phase

	Wang et al. [8]	Wang et al. [11]	Yuan, Yu [59]	Jiang et al. [28]
User	$2T_h+T_{GM}+2T_{GM}$ $+T^{d-1}_{Ge}+T_{BLS}$	$2T_h+T_{GM}$ $+T_{Ge}+T_{BLS}$	T_h+T_{GM} $+2T^k_{Ge}+T_{BLS}$	$T^{n-1}_{Ge}+T_{Ge}$ $+T_{GM}$
CSS	-	-	-	T_B
TPA	-	-	-	$T^{n-1}_{Ge}+6T_{Ge}+7T_{GM}+2T_h$

Table 8: Comparison of computation in user revocation phase

	Wang et al. [8]	Wang et al. [11]	Yuan, Yu [59]	Jiang et al. [28]
Data owner	No support	T_M	$3T_M+T_{Ge}$	-
User	No support	T_M	-	-
CSS	No support	$T_M+T_B+T_{BLS}$	T_{BLS}	-
TPA	No support	-	T_{GM}	$4T_{Ge}+2T_h+5T_{GM}+T_B$

Table 9: Comparison of storage

	Wang et al. [8]	Wang et al. [11]	Yuan, Yu [59]	Jiang et al. [28]
CSS	$IHT\{id,m,v,r\}$, σ	$IHT\{id,m,v,r,s\}$, σ	m, σ	aux, σ^t
TPA	No	No	B	RL

International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01), pp. 514–532, Gold Coast, Australia, 2001.

[7] D. Boneh, H. Shacham, "Group signatures with verifier-local revocation", in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pp. 168–177, Washington DC, USA, Oct. 25-29, 2004.

[8] W. Boyang, L. Baochun, "Oruta: Privacy-preserving public auditing for shared data in the cloud", *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 43–56, 2014.

[9] W. Boyang, L. Baochun, and L. Baochun, "Oruta: Privacy-preserving public auditing for shared data in the cloud", in *Proceedings of the 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS'13)*, pp. 93–98, Xian, China, Sept. 9-11, 2012.

[10] W. Boyang, L. Baochun, and L. Baochun, "Public auditing for shared data with efficient user revocation in the cloud", in *Proceedings of the 32nd IEEE International Conference on Computer Communications (INFOCOM'13)*, pp. 2904–2912, Turin, Italy, Apr. 14-19, 2013.

[11] W. Boyang, L. Baochun, and L. Baochun, "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.

[12] W. Boyang, L. Baochun, L. Baochun, and L. Fenghua, "Certificateless public auditing for data integrity in the cloud", in *Proceedings of the First IEEE Conference on Communications and Network Security (CNS'13)*, pp. 136–144, Maryland, USA, Oct. 14-16, 2013.

[13] W. Boyang, L. Baochun, and L. Ming, "Privacy-preserving public auditing for shared cloud data supporting group dynamics", in *Proceedings of IEEE International Conference on Communications (ICC'13)*, pp. 1946–1950, Budapest, Hungary, June 9-13, 2013.

[14] W. Boyang, L. Baochun, L. Xuefeng, L. Fenghua, L. Xiaoqing, "Efficient public verification on the integrity of multi-owner data in the cloud", *Journal of Communications and Networks*, vol. 16, no. 6, pp. 592–599, 2014.

[15] W. Boyang, S. S. M. Chow, L. Ming, L. Baochun, "Storing shared data on the cloud via security-mediator", in *Proceedings of the 33rd IEEE International Conference on Distributed Computing Systems (ICDCS'13)*, pp. 124–133, Pennsylvania, USA, July 8-11, 2013.

[16] D. Cash, A. Küpcü, D. Wichs, "Dynamic proofs of retrievability via oblivious RAM", in *Advances in Cryptology (EUROCRYPT'13)*, LNCS 7881, pp. 279–295, Springer, 2013.

[17] D. Catalano, D. Fiore, "Vector commitments and their applications", in *Proceedings of the 16th International Conference on Practice and Theory in Public-Key Cryptography (PKC'13)*, pp. 55–72, Nara, Japan, Feb. 26 - Mar. 1, 2013.

[18] L. Chang, et al., "Public auditing for big data storage in cloud computing – A survey", in *Proceedings of the 16th IEEE International Conference on Computational Science and Engineering (CSE'13)*, pp. 1128–1135, Sydney, Australia, Dec. 3-5, 2013.

[19] B. Chen, R. Curtmola, "Robust dynamic provable data possession", in *Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'12)*, pp. 515–525, Macau, China, 2012.

[20] L. Chen, "Using algebraic signatures to check data possession in cloud storage", *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1709–1715, 2013.

[21] W. Cong, W. Qian, R. Kui, C. Ning, L. Wenjing, "Toward secure and dependable storage services in cloud computing", *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 220–232, 2012.

[22] W. Cong, W. Qian, R. Kui, L. Wenjing, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing", in *Proceedings of the 29th IEEE Conference on Information Communications (INFOCOM'10)*, pp. 1–9, San Diego, California, USA, Mar. 14-19, 2010.

[23] C. Erway, A. K, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, pp. 213–222, Illinois, USA, 2009.

[24] S. Halevi, D. Harnik, B. Pinkas, A. Shulman-Peleg, "Proofs of ownership in remote storage systems", in *Proceedings of the Proceedings of the 18th ACM conference on Computer and Communications Security*, pp. 491–500, Chicago, Illinois, USA, 2011.

[25] C. Hanser, D. Slamanig, "Efficient simultaneous privately and publicly verifiable robust provable data possession from elliptic curves", *IACR Cryptology ePrint Archive*, pp. 392–406, 2013.

[26] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of big data on cloud computing: Review and open research issues," *Information Systems*, vol. 47, no. 6, pp. 98–115, 2015.

[27] W. Huaqun, "Proxy provable data possession in public clouds", *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 551–559, 2013.

[28] T. Jiang, X. Chen, J. Ma, "Public integrity auditing for shared dynamic cloud data with group user revocation", *IEEE Transactions on Computers*, to be published.

[29] Y. Jiawei, Y. Shucheng, "Secure and constant cost public cloud storage auditing with deduplication", in *Proceedings of the First IEEE Conference on Communications and Network Security (CNS'13)*, pp. 145–153, Maryland, USA, Oct. 14-16, 2013.

[30] Y. Jiawei, Y. Shucheng, "Efficient public integrity checking for cloud data sharing with multi-user mod-

ification", in *Proceedings of the 33rd IEEE International Conference on Computer Communications (INFOCOM'14)*, pp. 2121–2129, Toronto, Canada, Apr. 27 - May 2, 2014.

[31] L. Jin, T. Xiao, C. Xiaofeng, D. S. Wong, "An Efficient Proof of Retrievability with Public Auditing in Cloud Computing", in *5th International Conference on Intelligent Networking and Collaborative Systems (INCoS'13)*, pp. 93–98, 2013.

[32] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 584–597, Virginia, USA, 2007.

[33] Y. Kan, J. Xiaohua, "An efficient and secure dynamic auditing protocol for data storage in cloud computing", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717–1726, 2013.

[34] R. Kui, W. Cong, W. Qian, "Security challenges for the public cloud", *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[35] J. Li, X. Tan, X. Chen, D. Wong, and F. Xhafa, "OPoR: Enabling proof of retrievability in cloud computing with resource-constrained devices," accepted and to be publish in *IEEE Transactions on Cloud Computing*, Oct. 2014.

[36] C. Liu, J. L. Chen, T. Yang, X. Zhang, C. Yang, R. Ranjan, and K. Ramamohanarao, "Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2234–2244, 2014.

[37] C. Liu, R. Ranjan, C. Yang, L. Wang, and J. Chen, "MuR-DPA: Top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud", *IEEE Transactions on Computers*, to be published.

[38] C. C. Liu, X. Zhang, and J. Chen, "External integrity verification for outsourced big data in cloud and iot: A big picture," *Future Generation Computer Systems*, vol. 49, no. 6, pp. 58–67, 2015.

[39] S. Meena, E. Daniel, N. A. Vasanthi, "Survey on various data integrity attacks in cloud environment and the solutions", in *Proceedings of the 2013 IEEE International Conference on Circuits, Power and Computing Technologies (ICCPCT'13)*, pp. 1076–1081, Nagercoil, India, Mar. 20-21, 2013.

[40] P. M. Mell and T. Grance, *The NIST Definition of Cloud Computing*, Technical Report: SP 800-145, National Institute of Standards and Technology, 2011.

[41] R. C. Merkle, "Protocols for public key cryptosystems," in *IEEE Symposium on Security and Privacy*, pp. 122–134, California, USA, 1980.

[42] R. L. Rivest, A. Shamir, Y. Tauman, "How to leak a secret", in *Proceedings of the Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 612–613, Gold Coast, Australia, Dec. 9-13, 2001.

[43] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'08)*, pp 90–107, Melbourne, Australia, 2008.

[44] A. Shamir, "How to share a secret", *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[45] Y. J. Shin, J. Hur, K. Kim, "Security weakness in the proof of storage with deduplication", *IACR Cryptology ePrint Archive*, pp. 554, 2012.

[46] M. Sookhak, et al., "Remote data auditing in cloud computing environments: A survey, taxonomy, and open issues", *ACM Computing Surverys*, vol. 47, no. 4, pp. 1–34, 2015.

[47] M. Sookhak, H. Talebian, K. Ahmed, A. Gani, M. K. Khan, "A review on remote data auditing in single cloud server: Taxonomy and open issues", *Journal of Network and Computer Applications*, vol. 43, pp. 121–141, 2014.

[48] S. Subashini, V. Kavitha, "A survey on security issues in service delivery models of cloud computing", *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[49] B. Wang, B. Li, and H. Li, "Knox: Privacy-preserving auditing for shared data with large groups in the cloud", in *Proceedings of the Proceedings of the 10th International Conference on Applied Cryptography and Network Security (ACNS'12)*, pp. 507–525, Singapore, June 26-29, 2012.

[50] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[51] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in *Proceedings of the 17th International Workshop on Quality of Service (IWQoS'09)*, pp. 1–9, South Carolina, USA, 2009.

[52] Q. Wang, C. Wang, J. Li, K. Ren, W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing", in *Proceedings of the Proceedings of the 14th European Conference on Research in Computer Security*, pp. 355–370, Saint-Malo, France, Sept. 21-25, 2009.

[53] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

[54] L. Wei, et al., "Security and privacy for storage and computation in cloud computing", *Information Sciences*, vol. 258, pp. 371–386, 2014.

[55] Q. Wu, Y. Mu, W. Susilo, B. Qin, J. Domingo-Ferrer, "Asymmetric group key agreement", in *Advances in Cryptology (EUROCRYPT'09)*, LNCS 5479, pp. 153–170, Springer, 2009.

[56] Z. Yan, et al., "Dynamic audit services for outsourced storages in clouds", *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 227–238, 2013.

[57] Y. Yu, J. Ni, M. H. Au, Y. Mu, B. Wang, and H. Li, "On the security of a public auditing mechanism for shared cloud data service", *IEEE Transactions on Services Computing*, to be published.

[58] J. Yuan, and S. Yu, "Proofs of retrievability with public verifiability and constant communication cost in cloud", in *Proceedings of the Proceedings of the ACM International Workshop on Security in Cloud Computing (ASIACCS-SCC'13)*, pp. 19–26, Hangzhou, China, 2013.

[59] J. Yuan, and S. Yu, "Public integrity auditing for dynamic data sharing with multi-user modification", *IEEE Transactions on Information Forensics and Security*, to be published.

[60] Q. Zheng, S. Xu, "Secure and efficient proof of storage with deduplication", in *Proceedings of the Proceedings of the second ACM Conference on Data and Application Security and Privacy*, pp. 1–12, San Antonio, Texas, USA, Feb. 07-09, 2012.

[61] Q. Zheng, S. Xu, "Fair and dynamic proofs of retrievability", in *Proceedings of the Proceedings of the First ACM Conference on Data and Application Security and Privacy*, pp. 237–248, San Antonio, USA, 2011.

[62] Y. Zhu, H. Hu, G. J. Ahn, S. S. Yau, "Efficient audit service outsourcing for data integrity in clouds", *Journal of Systems and Software*, vol. 85, no. 5, pp. 1083–1095, 2012.

[63] D. Zissis, D. Lekkas, "Addressing cloud computing security issues", *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

Chih-Wei Liu received his M.S. in Soil And Water Conservation from National Chung Hsiung University, Taichung, Taiwan, ROC, in 2008. He is currently pursuing the Ph.D. degree from Computer Science and Information Engineering Department of Asia University, Taichung, Taiwan. His research interests include information security, cloud computing, and information law.

Wei-Fu Hsien received his B. S. in Department of Information Management from National Kaohsiung Marine University, Kaohsiung, Taiwan, ROC, in 2013. He is currently pursuing the M.S. degree with the Department of Management Information Systems from National Chung Hsing University. His research interests include security and privacy of cloud computing, and applied cryptography.

Chou-Chen Yang received his B.S. in Industrial Education from the National Kaohsiung Normal University, in 1980, and his M.S. in Electronic Technology from the Pittsburg State University, in 1986, and his Ph.D. in Computer Science from the University of North Texas, in 1994. From 1994 to 2004, Dr. Yang was an associate professor in the Department of Computer Science and Information Engineering, Chaoyang University of Tech- nology. Currently, he is a professor in the Department of Management Information Systems, National Chung Hshing University. His research interests include network security, mobile computing, and distributed system.

Min-Shiang Hwang received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China, in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a project leader for research in computer security at TL in July 1990. He obtained the 1997, 1998, and 1999 Distinguished Research Awards of the National Science Council of the Republic of China. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include database and data security, cryptography, image compression, and mobile communications.