Li, Angtai; Chen, Yu; Yan, Zheng; Zhou, Xiaokang; Shimizu, Shohei

A Survey on Integrity Auditing for Data Storage in Cloud: from Single Copy to Multiple Replicas

Published in:
IEEE Transactions on Big Data

DOI:
10.1109/TBDATA.2020.3029209

E-pub ahead of print: 01/01/2020

Document Version
Peer reviewed version

Please cite the original version:
Li, A., Chen, Y., Yan, Z., Zhou, X., & Shimizu, S. (2020). A Survey on Integrity Auditing for Data Storage in Cloud: from Single Copy to Multiple Replicas. *IEEE Transactions on Big Data*.
https://doi.org/10.1109/TBDATA.2020.3029209

The final version of record is available at http://dx.doi.org/10.1109/TBDATA.2020.3029209

A survey on integrity auditing for data storage in the cloud: from single copy to multiple replicas

Angtai Li, Yu Chen, Zheng Yan, ***Senior Member, IEEE,*** Xiaokang Zhou, and Shohei Shimizu

AbstractÑ The rapid advancement of cloud computing has promoted the development of cloud storage services. One of the biggest concerns of cloud users is whether the completeness and recoverability of data can be guaranteed when cloud servers encounter problems. Only when the integrity of data is fully guaranteed can users consume cloud storage with confidence, especially in a complicated cloud environment with multiple clouds. However, the literature still lacks a thorough survey on cloud data integrity auditing for both single copy and multiple replicas. In this paper, we survey and compare existing auditing schemes for single copy and multiple replicas based on a set of criteria. Based on our review and analysis, we discuss open issues, potential applications and future directions in the field of the integrity auditing in the cloud, including the implications of such trendy topics as merging blockchain and edge computing into data integrity auditing.

Index TermsÑ Integrity auditing, data storage, cloud computing, singly copy, multiple replicas

Ñ Ñ Ñ Ñ Ñ Ñ Ñ Ñ Ñ Ñ ! Ñ Ñ Ñ Ñ Ñ Ñ Ñ Ñ Ñ Ñ

1 Introduction

THe concept and model of cloud computing, which was formally proposed in 2006, has greatly fueled the development of the Internet industry and at the same time given rise to tremendous changes in social life. By using cloud storage services, which is built upon cloud computing, users can conveniently retrieve their own data at any time and any place, especially for users with limited local storage capacity to store and manage data. Cloud storage is becoming a trend by addressing the issue of users' local storage limitation [1].

However, despite the convenience brought by the cloud computing, security issues also rise [2]. For example, the cloud may disclose some information about the data, update the data, or even remove the data, whichh lead to distrust by the users. Thus, ensuring the integrity of the cloud data becomes essential [3]. To make this issue more complex, with the development of cloud computing, there may exist more than one single cloud server in a cloud system for the purpose of expanding cloud storage and computing ability. For example, the ZooKeeper, a distributed service framework proposed by Hunt et al. [82], uses multiple clouds to achieve the above goal. Built on top of ZooKeeper, DepSky, proposed by Bessani et al. [83], also deploys multiple and diverse clouds to form cloud-of-clouds. The enhanced multi-server architecture greatly increases the capabilities of the cloud. With an increasing number of architectures and applications built upon a multi-cloud environment, auditing data integrity becomes even more important as users need to interact with multiple clouds, a complicated environment that might include public clouds, private clouds or a combination of both.

The auditing data integrity in the cloud originates from the Remote Integrity Checking [12] in 2003. This was followed by a scheme of data integrity verification named Provable Data Possession (PDP) [14] in 2007, the basic of many variants of the PDP scheme afterwards. At the same time in 2007, the scheme of Proof of Retrievability (POR) [42] was officially proposed. Unlike PDP, POR focuses on data recovery when the data is verified as incomplete. Both PDP and POR are the earliest solutions of data integrity auditing in a single cloud environment, in which only one copy of user data is stored in the cloud. Afterwards, with the development of the multi-cloud architecture, variants of PDP and POR for multi-cloud environments have appeared on the basis of both of them. In this scenario, users who store multiple replicas of their data (in multiple clouds) would have to audit all of replicas.

Prior work has presented systematic reviews on integrity auditing, but primarily focused on a single-cloud scenario, while a comprehensive survey on integrity auditing

¥ *A. Li is with School of Cyber Engineering, Xidian University, Xi'an 710071, China.*
E-mail: angtai123@163.com
¥ *Y. Chen is with School of Information Systems and Technology, San Jose State University, San Jose, CA 95192, USA.*
E-mail: yu.chen@sjsu.edu
¥ *Z. Yan is with the State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an 710071, China and with Department of Communications and Networking, Aalto University, Espoo 02150, Finland.*
E-mail: zyan@xidian.edu.cn
¥ *X. Zhou is with School of Cyber Engineering, Xidian University, Xi'an 710071, China.*
E-mail: zyan@xidian.edu.cn
¥ **S. Shimizu** *is with School of Cyber Engineering, Xidian University, Xi'an 710071, China.*
E-mail: zyan@xidian.edu.cn
*****Please provide a complete mailing address for each author, as this is the address the 10 complimentary reprints of your paper will be sent**

*Please note that all acknowledgments should be placed at the end of the paper, before the bibliography (***note that corresponding authorship is not noted in affiliation box, but in acknowledgment section***).*

TABLE 1
COMPARISON WITH EXISTING SURVEYS

Topic	[4]	[9]	[5]	[8]	[6]	[10]	[7]	[74]	[75]	[76]	[77]	Our Survey
Give a review of Provable Data Possession (PDP)	No	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes
Give a review of Proof of Retrievability (POR)	No	Yes	No	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes
Compare PDP and POR	Yes	No	Yes	Yes	Yes	No	No	No	No	No	Yes	Yes
Give a comprehensive review of auditing schemes (including PDP and POR)	No	No	No	No	No	No	No	No	No	No	No	Yes
Focus on auditing on single copy and multiple replicas	No	No	No	No	No	No	No	No	No	No	No	Yes

Yes: including; No: not including.

for data storage in multiple clouds still lacks. For instance, Chen et al. [4] and Qin et al. [5] focused on the classification of PDP and POR in the single cloud, but ignored the auditing work of multiple replicas, similarly to the surveys performed by Hsien et al. [6] and Thangavel et al. [75]. A comprehensive review by Tan et al. [7] conducted a comprehensive review only on the schemes of POR. On the other hand, Barsoum [74] and another one by Dong et al. [76] mainly focused on the schemes of PDP, and yet little attention paid to multiple replicas in multiple clouds. Sookhak et al. [8] presented a more comprehensive survey on both PDP and POR, but only focused on the single cloud scenario. Azain et al. [9] and Wang et al. [10] discussed data security issues from a single cloud to multiple clouds, but their studies lack details on the data storage integrity auditing. Although Zhou et al. [77] covered both single copy and multiple replica, they did not distinguish PDP from POR, which is addressed by our paper. Table 1 summarizes the differences of our survey from existing ones.

In this paper, we present a detailed review on the integrity auditing on both single copy and multiple replicas from a single cloud to multiple clouds. We survey both the PDP and the POR schemes used for single copy auditing. Our review employs a set of evaluation criteria, which is originally proposed by us to measure the pros and cons of each existing work, and further assist us to figure out open issues and interesting research directions in this field. Furthermore, as cloud computing also involves distributed computing and Internet of Things (IoT), we also discuss how our findings might be merged into related trendy topics such as blockchain and edge computing.

Therefore, the contributions of this survey can be summarized as follows:

¥ A comprehensive literature review about data integrity auditing, including auditing on a single copy and multiple replicas of data;

¥ Open research issues and challenges in integrity auditing based on the literature review, comparison and discussion;

¥ Future directions about auditing work in the cloud and its applications in blockchain and edge computing.

The rest of this article is organized as follows. Section 2 presents preliminaries of the data auditing in the cloud and related concepts. Section 3 presents the criteria for evaluating data auditing schemes. Section 4 reviews and compares various work about auditing in detail in different cloud environments. This is followed by open issues and proposed future research directions in Section 5. Finally, the paper concludes in the last section.

2 PRELIMINARIES

Before presenting the review of specific schemes, we provide the preliminaries about the basic auditing single copy in the single cloud (Section 2.1) and multiple replicas in multiple clouds (Section 2.2), followed by a discussion about its relationship with blockchain technology.

2.1 Auditing on Single Copy in a Center Cloud

With the increasing amount of data generated by users across various devices (e.g., smart phones, tables, IoT devices) and limited user local storage space, there is an increasing demand to outsource user data to the cloud for management. In the traditional cloud computing model, only one copy needs to be stored in the cloud server. When users need their data, the data is retrieved from one single cloud server. However, once users store data in the cloud, they also lose the full control of their own data. Therefore, it is imperative to guarantee users' data integrity, a main goal of data auditing. Auditing might be conducted by users themselves, but when users cannot complete the auditing work due to computing power or some other resource constraints, auditing can be conducted by the third auditors with users' authorization.

Current work on data auditing in the cloud mainly fall into two categories: Provable Data Possession (PDP) and Proof of Retrievability (POR). In a typical cloud storage scenario, users upload their data to the cloud, then perhaps delete the locally stored files due to local storage constraints, and then all management on the data is performed in the cloud. In order to ensure the completeness of data, the traditional solution is that users download the data from the cloud and upload again after checking the completeness. However, this method is unlikely to be impractical, typically requiring huge communication overhead between the users and the cloud. Another solution is that the cloud checks the data integrity without the users downloading the data. The figure illustrates the PDP scheme, in which the users challenge the cloud and the cloud sends proof of the data integrity. The PDP scheme can detect the completeness of data, but destroyed data cannot be recovered. The Proof of Retrievability (POR) scheme, however, is designed to ensure destroyed data can be recovered afterwards. We summarize the auditing process is shown in Fig.1.

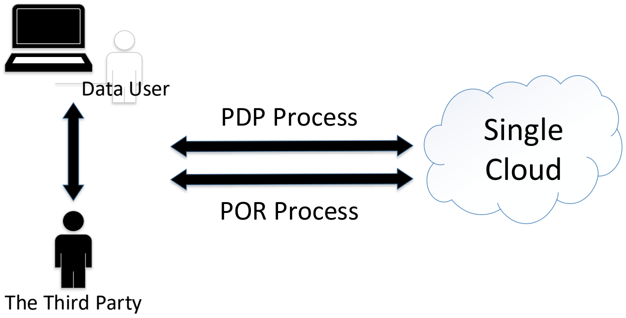

Fig. 1. Single Copy Single Cloud Auditing Model. The integrity auditing process between a user and a single cloud, including the schemes of PDP and POR. The third party can assist the user to finish the auditing process.

2.2 Multiple Replicas and Distributed Auditing

However, POR cannot solve all problems. For example, when the cloud server experienced some irreversible problems, such as physical equipment damage, the data is permanently unable to recover even under POR. To address this issue, researchers then proposed storing multiple replicas of a file on multiple servers in a distributed system. In this case, even if a copy of the file on one server is damaged, users can recover the file from other servers.

Another motivation of using multiple cloud is that single replica single cloud used in the traditional cloud computing can no longer meet the needs of the current mainstream distributed computing model. In cloud computing, distributed computing spreads the computational power of a cloud data center to multiple clouds. Multiple clouds can include either public clouds or private clouds, or a mixed of both. Since the process of cloud storage and computing is coordinated by multiple clouds, it can also be regarded as one distributed system. This is particularly useful for tasks that require a lot of computing power or other computing resource. Therefore, the model of distributed computing might be better at leveraging computing resources and improving efficiency compared to the simple cloud model. Applying distributed to cloud computing actually weakens the computing power of cloud data centers, and the network central nodes are no longer needed because the processing of data is distributed to multiple servers.

When a user's files are stored in a distributed system, (called a file partition), these files are quickly duplicated and distributed. Typically, multiple copies are generated for each file. The existence of multiple servers may require private communication among servers and prevent any deception about file storage. To store and access the files, the user may need to interact with multiple servers, which imposes requirements towards user's resources, such as storage and computing power to be improved. Meanwhile, similarly to single copy single cloud model, users could leverage a trusted third party to reduce their computational overhead. The distributed model, or multiple replica multiple cloud model, can be summarized in Fig.2.

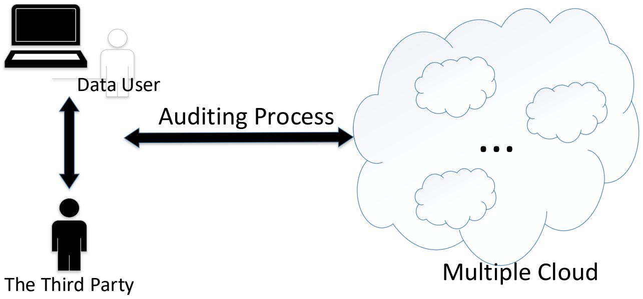

Fig. 2. Distributed Model or Multiple Replica Multiple Cloud Model. The integrity auditing process between a user and multiple clouds in a distributed system. The third party can help the data user to finish the auditing process. The auditing process includes PDP and POR as in Fig1.

The two models summarized in Fig. 1 and Fig. 2 Ð single copy auditing in a single cloud and multiple replicas auditing in multiple clouds (in a distributed way) are the two basic audting contexts reviews in this paper.

3 CRITERIA OF AUDITING

3.1 Evaluation Criteria

In this section, we list a number of criteria for evaluating integrity auditing schemes for both single copy and multiple replicas. These criteria are summarized based on the current surveys as listed in Table 2, where we indicate whether the existing surveys have considered the criteria. In Table 2, a blank indicates that a criterion was not considered.

The criteria that we apply in our survey are summarized and explained as follows:

1) Storage overhead
2) Communication overhead
3) Computational overhead
4) Dynamic data operation
5) Number of verifications
6) Public auditing
7) Bach auditing
8) Privacy protection
9) Provable security
10) Retrievability

We then explain the criteria below.

1) *Storage overhead* refers to the storage cost of the cloud and the cloud user. Storage constraints is one of the major reasons for users to store their data in the cloud, so it is crucial to consider user storage cost during auditing that should not introduce much extra storage cost to the user.

2) *Communication overhead* refers to the overhead that occurs during the integrity verification process when a user needs to interact with the cloud. For example, during the auditing process, the user sends a challenge to the cloud, and the cloud responds by proving the integrity status of their data. Thus, it is necessary to consider data transmission overhead during the above interaction process.

3) *Computational overhead* occurs when distributing the

verification proof, during which the cloud generates label tags and sends them to the user based on the data the user stored. Sometimes the user side also needs to perform some computation during the verification process. Thus, computational overhead shows the efficiency of the data integrity auditing.

4) *Dynamic data operation* refers to the data integrity auditing should support procedure after the users store the data in cloud for data integrity. Such operation includes adding or deleting data, which commonly happens in file and data management.

5) *Unlimitation of verification* refers to that the times that the users can request the cloud for verification is unlimited, which is also a standard for evaluating the quality of an auditing scheme.

TABLE 2
CRITERIA IN EXISTING SURVEYS

	Storage over-head	Communica-tion over-head	Computa-tional over-head	Dynamic data opera-tion	Unlimita-tion of veri-fication	Public auditing	Batch auditing	Privacy preserve	Provable security	Retrieva-bility
[4]	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
[5]	No	Yes	Yes	Yes	No	No	Yes	Yes	No	No
[6]	Yes	No	Yes	Yes	No	No	Yes	Yes	No	No
[7]	Yes	Yes	Yes	Yes	No	No	No	Yes	No	No
[8]	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	No
[9]										
[10]	No	No	No	Yes	No	No	Yes	No	No	No
[74]	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
[75]	No	No	No	Yes	No	No	Yes	No	No	Yes
[76]	Yes	No	No	Yes	No	Yes	Yes	Yes	No	No
[77]	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	Yes

6) *Public auditing* refers to an auditing work is handled by (trusted) third parties. This is essential when user computing capacity is limited to conduct auditing locally.

7) *Batching auditing* aims to audit data integrity for multiple users simultaneously. This feature could greatly improve the efficiency of auditing; thus, it is an advantage if a scheme has such a capability.

8) *Privacy protection* aims at protecting user privacy in the auditing process. The cloud or the third party that conduct the auditing may get partial or all of the user data. So, it is essential to preserve privacy in order not to reveal any information about users to any third parties [11].

9) *Provable security* refers to the level of security that an auditing scheme can achieve, measured through models in cryptography. A secure auditing scheme is usually designed with cryptographic tools. Whether a scheme can be safe under a standard model or random Oracle model is also an important criterion for evaluating the schemes.

10) *Retrievability* means that the cloud should prove to the users that they can retrieve the initial data. Even if the data is damaged, the users can recover their complete data based on the metadata.

In Section 4, we compared POP and POR in a single cloud and multiple replicas auditing in multiple clouds. When it comes to a distributed system and user data are stored in many cloud servers, the ability to verify which server occurs an error during data storage and data processing is also a key criterion. More importantly, when storing data in a distributed system, multiple replicas can be generated from a single copy of a file and they are stored in different servers. These servers may reside in different geographic locations. Under such a condition, when an error occurs, the location of the server where the file is damaged should be tracked by the auditing schemes. Thus, we propose a new criterion named "Error Tracking" to judge whether this feature can be supported in auditing.

3.2 Discussion

In Section 4, we will comprehensively review the existing work about data integrity auditing and provide Table 3-5 to compare reviewed schemes. In our review, we employ the above proposed criteria to judge the performance of existing work. For Criteria 1 to 3, they are all about efficiency. Because there is not a good standard to uniform efficiency, we use low storage, and high computation & communication efficiency to simply evaluate the existing schemes. For the remaining seven criteria, we use Yes or No to show that whether a scheme can functionally support the criteria (i.e., the expected features) or not. What is more, for the one criterion that are only applied to evaluate the auditing scheme for multiple replicas, we discuss it in Section 4.3 (Auditing Schemes for Multiple Replicas). Except the criterion ÒError TrackingÓ, we treat other criteria equally in both the single copy auditing and the multiple replicas auditing. In addition, during the paper review, we judge why the contents of Table 3-5 should be marked like that. In what follows, we specify the common annotations used in Table 3-5.

•Low Storage

- Yes: The storage cost of a data owner (i.e., a cloud user) or the cloud is relatively lower than the cost of other schemes.
- No: The storage cost cannot reach a low level or a scheme has a high storage cost in order to achieve another criterion.

•Communication and computation efficiency

- Yes: The efficiency of communication or computation at the side of a data owner or the cloud is high based on experimental tests or the efficiency is improved compared with other existing schemes.
- No: The efficiency is not improved or is low.

•Dynamic data operation

- Yes: A scheme supports dynamic operation on data during auditing. The dynamic operation can be all types of dynamic operations or some of them.
- No: Dynamic operation cannot be supported by a scheme.

•Unlimitation of Verification

- Yes: This means the times that a user can ask the cloud to give the proof is unlimited.
- No: This means that the times that a user can ask the cloud to verify the integrity is limited.

•Public auditing

- Yes: In a scheme, a third party except for a data owner can ask the cloud to provide a proof of data integrity. The data owner can outsource the auditing work to the third party or the third party can help the data owner perform auditing.
- No: This means a scheme cannot support the third party to do the auditing.

•Batching auditing

- Yes: A scheme supports auditing many files at one time in order to save time and improve efficiency.
- No: Only one data auditing work can be run in one time.

•Privacy preservation

- Yes: During auditing, no one can get the knowledge of a data owner, which means the auditing protects the privacy of the owner.
- No: A scheme could disclose some knowledge of the data owner. It also means that the scheme cannot protect data owner privacy.

•Provable security

- Yes: A scheme provides a security proof under one of the following models: Random Oracle model or a standard model.
- No: There is no security proof provided or a scheme has been proved as unsafe.

•Retrievability

- Yes: This means that a scheme can prove to a data owner that its file can be withdrawn from the cloud.
- No: A scheme does not support this property.

•Error tracking

- Yes: Under the multiple replicas model, a scheme can distinguish which copy is destroyed.
- No: A scheme cannot support error tracking or it cannot track an error file.

Notably, retrievability can only be supported by POR schemes, while cannot be offered by PDP schemes. For auditing multiple replicas, most of existing schemes are based on PDP. So, we do not put Òretrievability Ó in Table 3 and Table 5. For multiple replicas, as we will show in Table 5, one important criterion is Òerror trackingÓ for distinguishing which file goes wrong in which cloud server. But for an efficiency and secure scheme for auditing data integrity, all of the proposed criteria should be considered.

4 LITERATURE REVIEW

Based on the above criteria, we conduct a comprehensive review on three types of auditing schemes: schemes based on PDP, schemes based on POR and schemes for multiple replicas. In particular, we separately review the schemes for dynamic data operation and public auditing in the in single copy auditing, which are highly researched in the current literature. Specifically, we review the schemes with error tracking in the part of multiple replicas auditing in Section 4.3.1.

4.1 Auditing Schemes Based on PDP

Before PDP was formally defined, Deswarte et al. [12] first discussed the integrity check of remotely stored data. They used an RSA-based hash function to calculate the hash value for the entire file and add a challenge when calculating the checksum of a file. This method was proposed based on public key cryptography and a verification protocol, which was built on the basis of Diffie-Hellman key exchange. Similarly, Filho and Barret [13] proposed a data possession protocol based on the RSA-based secure hash function, with the purpose of preventing spoofing in data transmission. However, it is obvious that the computational cost of the scheme was very high. Since these two represent the earliest auditing scheme of auditing, the other criteria we proposed were not discussed.

The first formal definition of the PDP scheme was proposed by Ateniese et al. [14]. The two PDP schemes in their article used homomorphic verifiable tags. The user generated a tag for each data block, and stored the tag together with the data on the server. At the time of verification, the user randomly selected some blocks to challenge the server and asked the server to return the evidence of holding the data. The server used the request block and the corresponding tag to generate the possession evidence. Because of the homomorphism [15], the tags of multiple file blocks could be aggregated into one value, thus greatly saving the response bandwidth. The user confirmed the data possession by verifying the response information without retrieving the data. The proposed scheme only needed the user to maintain the constant metadata information, without the server needing to access the entire file. Experiments showed that the performance of their scheme was limited by disk I/O and not by cryptographic calculation. Their scheme provided a probabilistic proof and also supported public verifiability. The number of times the client could challenge the server to prove data possession had no restriction. However, since the scheme used RSA-based modular operations when generating evidence, it did not support dynamic data operations. Storage and computing efficient were not mentioned in their article; nor did batch auditing, privacy and security.

Later, Sebe et al. [16] presented a new protocol, adding a probabilistic optimization described in Ateniese et al [3], their scheme also did not need to store all the data files, just requiring the summary at the verifier. Even if there exists a person who was deceiving or an untrusted channel between the verifier and the server, the scheme was safe. And a balance could be reached between the cloud's computation overhead and the verifier's storage overhead. Communication overhead was reduced and the number of verified times was not limited.

Ateniese et al. [18] then propose a general mechanism for constructing a public key homomorphic linear authenticator (HLA) using any identifiable identities in the Random Oracle Model and showing how to convert any public key HLA into a public verifiable storage scheme, making communication complexity independent from file length and supporting unlimited verification. However, the storage efficiency was not clearly evaluated. Additionally, since the scheme was also based on public key cryptography, the efficiency might be a bit poor and lack the consideration about dynamic data operation. Furthermore, batch auditing, privacy and security issues were not mentioned yet.

Based on the work of Wang et al. [22], Yang et al. [24] discussed how auditing can be done on resource-constrained mobile devices, outsourcing users' computing tasks to a third-party auditor. They used bilinear signature and Merkle hash tree (MHT), which could help reduce communication and storage burden and also support dynamic data update. In their scheme, users only needed to undertake a small amount of computing work such as generating some private keys and random numbers, with the help of a trusted platform. Nevertheless, other criteria we proposed were not mentioned in this paper.

Wang et al. [28] presented the first ID-RDPC (identity-based remote data possession checking) protocol, in order to solve the problem of considerable costs in validating the usersÕ certificates designed in the PKI (public key infrastructure). Later Wang et al. [28] introduced identity authentication, a novel agent-oriented data upload and identity-based public remote data integrity check model with key encryption in the public cloud. The proposed protocol could prove to be secure and was based on the computation of the Diffie-Hellman problem. Based on the users' authorization, the protocol enabled private remote data integrity checking, trusted remote data integrity checking, and public remote data integrity checking. This work, however, paid little attention to public auditing, batch auditing and provable security.

Yang et al. [30] improved a new scheme on the basis of the PANDA scheme [29]. PANDA could not retain the shared data privacy in cloud storage and was vulnerable to complete forgery attacks, which might be performed by malicious cloud servers. Without proper data storage, valid auditing certificates could be falsified for any auditing. Then they proposed PANDA's improvement scheme supporting data privacy protection and public auditing, while generating the best communication and computational overhead. This paper does not cover the criteria of storage overhead, dynamic operation, number of verification and security proof.

The identity-based auditing scheme was also proposed in the work of Li et al. [34], which used a fuzzy identity-based data integrity auditing solution that simplified key management where a userÕs identity could be viewed as a set of descriptive attributes. A new primitive soundness was formally formalizing in the system model and the security model. Their scheme revolutionized key management in traditional remote data integrity checking.

A further article by Liu et al. [35] proposed a fine-grained data integrity auditing scheme. First, it supported update operations based on bilinear signature scheme and Merkle hash tree construction. Also, there was no limit to the size of file blocks, and public auditing was supported. In addition, their scheme had better security and could prevent attacks from malicious attackers and untrusted third-party auditors. Their solution effectively reduced communication overhead, but only considered the computational overhead and public auditing.

Traditional integrity auditing techniques used a cryptographic tool of hash algorithm, but most hash algorithms were vulnerable to third-party attacks. Traditional encryption algorithms such as Advanced Encryption Standard (AES), Fully Homogeneous Encryption (FHABE), and Keyword Policy-Based Encryption (KP-ABE) had failed due to the constraints of computing resources and memory. Therefore, Kalangi et al. [38] proposed a novel multiple-user ciphertext policy attribute based on integrity verification and encryption (MFM-CP-ABE) model. The MFM-CP-ABE model calculated the integrity value by treating the fingerprints of multiple users as attributes of encryption. The model was a combined integrity method for multiple-user fingerprint detail (MFM) extraction strategy and an improved ciphertext policy attribute-based encryption (ICP-ABE) algorithm. This model was effective compared to traditional models in terms of encryption and decryption time and data size. However, they only mentioned the criterion of computational efficiency in their article.

Nayak et al. [39] further proposed a secure and effective privacy protection provable data possession scheme (SEPDP). SEPDP was also extended to support multiple owners, data dynamics and batch auditing, while other criteria were not mentioned. The most attractive feature of the scheme was that auditors could verify the data with low calculation overhead and enhanced storage efficient. However, the paper by Yu et al. [40] showed that this solution did not guarantee fundamental security and that a malicious cloud could generate proofs to pass the verification by a third-party auditor even if it did not store the user's entire file.

Li et al. [41] proposed a new approach of PDP approach with lower client cost due to the constant amount of metadata. Based on bilinear groups, they proposed simple and effective auditing services supporting public auditing and untrusted outsourced storage. The goal was to solve the client's metadata by considering the cost of generating the verification, but the number of verifications was not mentioned in their article. In addition, their approach supported data dynamics and public verifiability. Extensive experimental results demonstrated its high efficiency. But it hardly considered batch auditing, privacy preservation and security proof.

We list the papers under the two criteria. For the two criteria, we think that they are the main point in the PDP research process.

4.1.1 Dynamic Data Operations of PDP

Based on the PDP scheme, Ateniese and Pietro et al. [17] improved the basic scheme and proposed an efficient and secure PDP scheme. They used symmetric cryptography and security could be proved under the Random Oracle Model, included dynamic data operations, i.e., modifications on the data such as deleting and appending. When the scheme was initialized, the user set the number and content of the challenge, and stored the response as metadata. Therefore, the number of challenging was limited. Because of using symmetric cryptography, public auditing was not supported in their scheme. Despite of the improved model, the paper placed little emphasis on storage and computation efficiency and batch auditing and privacy were not involved either.

Erway et al. [19] proposed two dynamic PDP schemes in their article. They implement a PDP scheme that supported all dynamic data operations for the first time: one used a level-based authentication jump table and the other was based on the RSA tree structure. The main goal was to support dynamic, insertion in particular. Their schemes were based on a new variant of authenticated dictionaries, using rank information to organize dictionary entries. The entire scheme was still based on RSA's modular operation and the price of dynamic performance changed from O (1) to O (log- n) (or O (n log-n)) for a file consisting of n blocks. So, their scheme was with high cloud server computation. This paper only discussed about communication efficiency and dynamic operation.

Later, Gritti et al. [25] also proposed an auditing scheme supporting data privacy and public auditing, which was more efficient in practice for computing and communication overhead. Asymmetric pairing was used to increase efficiency, and their scheme also supported dynamic operations. Comparing to previous works, their scheme demonstrated improved the efficiency. In this work, storage efficiency, number of verifications, batch auditing and provable security were not discussed.

In the paper of Zhang et al. [86], they develop a novel and efficient scheme using a technology called balanced update tree. In terms of the criteria we have listed, they focused on the data update operation. The computation and communication overhead of their scheme was lower than those of the state-of-the-art schemes.

In the scheme of Yao et al. [87], they proposed a new dynamic PDP scheme by introducing a secure signature scheme and the Large Branching Tree (LBT). Their scheme supported fully dynamic updates including modification, insertion and deletion. By replacing Merkle Hash Tree (MHT) with LBT, they achieve better efficiency of less communication cost. They employ a secure signature algorithm which greatly reduced computation cost both on CSP and client. What's more, they proved the security of their scheme.

4.1.2 Public Auditing of PDP

Shah et al. [20] first introduced the third-party auditors in their article, introducing external auditing and internal auditing. In a follow-up scheme, Shah et al. [21] allowed third-party auditors to periodically verify the data and return the result to the user in the scheme, thus reducing the user's verification burden and supporting privacy protection without revealing the user’s data. That was because the auditors had no knowledge of users' data or content. But the scheme required the trustworthiness of third-party auditors no conflicts with other entities. Additionally, their protocols relied upon computationally hard problem and no party could efficiently solve these problems. This work, however, did not cover the storage efficiency aspect.

Similarly, the scheme of Wang et al. [22] also allowed third-party auditors to verify the integrity of stored data in the cloud on behalf of users and supported dynamic data operation. It improved existing storage models by using classic Merkle hash tree and the technique of bilinear aggregate signature to handle multiple auditing tasks and construct multiple user setting. Thus, the scheme enabled the third-party auditors to perform multiple auditing tasks simultaneously, and thus enhancing efficient of the dynamic data operations. The limitations, however, was the lack of emphasis on the storage efficiency, number of verification times, batch auditing and security.

To protect user data from third-party auditors, Wang et al. [23] proposed a public auditing scheme in their article that supporting privacy protection and the auditors could not know the user's data information. Their scheme was the first one proposed to support scalable and efficient public auditing in cloud computing. Their work also supported multiple auditing simultaneously conducted by auditors. The scheme used public key based on homomorphic authenticator and random masking. They evaluated the safety and performance. However, storage efficiency, dynamic operation, verification number and batch auditing were also not discussed in their schemes. Gritti et al. [25] also discussed public auditing.

Similarly, Wang [26] studied proxy-provable data possession in his paper which could be thought as public auditing. In a public cloud environment, proxy proof was important when users are unable to perform remote data auditing. Based on the bilinear pairing technique, a reliable and efficient secure-provable scheme was designed, including the security and performance analysis, while other criteria were not mentioned.

Following this line of work, Wang et al. [29] proposed a novel public auditing mechanism (PANDA) that took the integrity of effective user revocation and shared data into account. Using the idea of proxy re-signing, the cloud was allowed to re-sign the block on behalf of an existing user during user revocation so that existing users did not have to download and re-sign the block. In addition, public verifiers were always able to audit the integrity of shared data without having to retrieve the entire data from the cloud, even if some portion of the shared data had been re-signed by the cloud. This mechanism could support batch auditing, that is, verifying multiple auditing tasks at the same time. Experimental results showed that their mechanism could significantly improve the efficiency of user revocation. However, their scheme only focuses on public auditing and batch auditing, and their scheme was later improved unsafe by Yang et al. [30].

Yuan et al. [31] also proposed a new integrity auditing scheme for cloud data sharing services, allowing multiple-user modification and public auditing, with high error detection probability and effective user revocation with practical computing communication auditing performance. Their solution was the first to protect against user impersonation attacks. The scheme also advanced previous schemes by supporting batch auditing of multiple tasks. However, computation efficiency, public auditing and privacy preservation were discussed in the article.

Similarly, Zhang and Dong [32] proposed an ID-based efficient cloud data integrity auditing protocol with provable security and rigorous security protocol proof under Random Oracle model. They extend the scheme to support multiple-user batch auditing. Their auditing protocol was demonstrated to be safe and effective, especially in a multiple-user environment, reducing the auditing staff's computing costs of auditing. However, then the insecurity of this scheme was pointed out in [33]. The other criteria we proposed were not involved in their scheme.

Jiang et al. [36] further proposed a public auditing scheme for shared data that supported group users to revoke. They studied a secure and efficient shared data integrity auditing protocol that supported multiple-user operations on an encrypted database. By using asymmetric group key exchange and group signature technology, the scheme incorporated an efficient scheme with some new features such as traceable and countability. They also provided an analysis of the safety and efficiency of the scheme. Their scheme focuses exclusively on the criteria of public auditing and security proof.

Shen et al. [37] proposed an efficient public auditing protocol with a new dynamic structure, combining bidirectional link information tables and position arrays. This structure successfully handled the relationship between a

TABLE 3
COMPARISON OF PDP SCHEMES AND PDP-BASED SCHEMES

	Low storage	Communication and computation efficiency	Dynamic data operation	Unlimitation of verification	Public auditing	Batch auditing	Privacy preservation	Provable security	Retrievability
[12]		No							No
[14]			No	No	Yes				No
[17]			Yes	Yes	No			Yes	No
[18]		Yes	No	No	Yes				No
[19]		No	Yes						No
[20]		No			Yes		Yes		No
[22]		Yes	Yes		Yes		No		No
[24]		Yes				Yes	Yes	Yes	No
[23]	Yes		Yes		Yes				No
[25]		Yes	Yes		Yes		Yes		No
[26]		Yes	Yes		Yes			Yes	No
[28]					Yes	Yes		No	No
[29]		Yes			Yes		Yes		No
[31]		Yes			Yes	Yes	Yes		No
[32]						Yes		No	No
[35]		Yes			Yes				No
[36]					Yes			Yes	No
[37]			Yes		Yes	Yes			No
[38]		Yes							No
[39]		Yes	Yes			Yes	Yes		No
[41]	Yes	Yes	Yes		Yes				No

Yes: supporting; No: not supporting; Blank: not mentioned.

given block of data and its specific location, making data updating and batch auditing more convenient. Therefore, while reducing data freshness, supporting delayed updates reduced overhead. Their protocol supported a variety of auditing attributes, which made the scheme more practical. Public auditing was supported by a trusted TPA that was equipped with expertise to ease the burden on users. The scheme supported batch auditing, which saved time and resources when verifying multiple data files simultaneously. Nevertheless, other criteria were not mentioned in their scheme.

We give a comparison of PDP schemes in Table 3.

4.2 Auditing Schemes Based on POR

The PDP schemes only check whether the data stored in the cloud was complete or not, but could not support the retrievability of the data. While, POR schemes were designed to address the retrievability issue.

Juesl et al. [42] first proposed the concept of POR and proposed a scheme based on "sentinel". The basic idea was to encrypt the file and use the error correction code to encode, and randomly insert the "sentinel" which was indistinguishable from the file data in the encoded file, requiring the inspector to ask the server to return the "sentinel" at these random positions during the challenge. The file was proved recoverable as long as receiving a respond from the server with a probability greater than a certain value. Because every time a challenge consumed a sentry, there was no challenge update mechanism, and only a limited number of challenges could be made. According to the criteria we proposed, they only mentioned the verification times.

Dodis et al. [44] first proposed and theoretically analyzed the POR code and provided several methods to converting the POR code into the POR scheme. They proposed a trade-off between security and other performance (e.g. auditing times, challenge locations, and server storage overhead), but the communication and computational overhead are not specifically considered in their paper, and the data update problem was not considered either.

Bowers et al. of RSA Lab [45] also proposed a theoretical framework for designing POR to reduce storage overhead and error detection rate. They identified file updates and public auditing as unresolved issues, but other criteria were not involved in their paper.

Armknecht et al. [53] proposed a novel storage efficiency POR, called SPORT, which transparently supported multiple-tenancy and deduplication. More specifically, SPORT enabled tenants to securely share the same POR tag to verify the integrity of the deduplication files, and thus greatly reducing the storage overhead. Cloud storage providers deducted the same content when storing tags for different tenants. SPORT was shown to be able to resist-malicious cloud providers and collusion between the tenants and the cloud part. The evaluation on their prototype indicated the proposal scheme generated tolerable computational overhead tenants for cloud providers.

Chavan et al. [54] further studied data storage in cloud computing to ensure data integrity. The computational cost of the user side during the reduction of integrity verified their data, and the concept of the public had been proposed to be verifiable. Their approach was to create a new entity named Cloud Service Controller (CSC) to reduce the trust party auditor (TPA) to third parties. The security was enhanced by modeling generations using AES encryption with SHA-512 and tags.

4.2.1 Dynamic Data Operation of POR

In this subsection, we review the existing work of POR that can support dynamic data operation.

In the case of the need to safely manage dynamic data, Zheng et al. [46] proposed the first dynamic POR scheme. Introducing a new POR scheme, called fairness, which was also inherent in dynamic data setup. Because of uncertainty, dishonest customers could legitimately use honest cloud storage servers to manipulate their data. Their solution was based on two new tools, one is the authentication data structure, called range-based 2-3 tree (rb23Tree) in short, and the other is the incremental signature scheme called hash-compress-And-sign (HCS). In this paper, support on dynamic operation is the focus.

Further, Mo et al. [79] focused on the efficiency problem. They extended the static POR scheme to a dynamic one, in which a client can perform update operation such as insertion, deletion and modification. They developed a new version of authenticated data structure based on B+ tree and a Merkle hash tree and named it Cloud Merkle B+ tree (CMBT). By combining the CMBT with the BLS signature, they proposed a dynamic POR scheme. Their scheme's worst communication complexity is O(logn).

Similarly Cash et al. [49] proposed a POR scheme for dynamic storage, allowing clients to perform arbitrary reads and writes by running an efficient protocol at any location of a server and perform auditing protocol to ensure that the server to maintain the latest version of the data. Computation and communication complexity in their protocol were just multiple-log of data size. Their main idea was dividing the data into small blocks and individually encoding each block in a redundant manner so that updating the data block in any internal way could only affect some code character numbers. Computation and communication complexity in their protocol were greatly reduced. Except for computation efficiency and dynamic operation support, other criteria were not considered.

Shi et al. [78] also discussed efficiency in their dynamic POR scheme with constant client storage by using homomorphic checksums. They also showed how to make their scheme publicly verifiable. However, the other criteria as we list in Section 3 were not mentioned in their work.

Later Li et al. [48] proposed a new cloud storage solution called OPoR, which enhanced the POR model to support dynamic data operations and resist against reset attacks initiated by the cloud storage server during the upload phase. OPoR outsources heavy calculations of tag generation to the cloud auditing server and eliminates usersȭ involvement in the auditing and pre-processing phases. This article, however, only covers dynamic and security.

4.2.2 Public Auditing of POR

To reduce the computational cost of the user side during integrity auditing, public auditing emerged as a solution. For example, as discussed in 4.2.1, the scheme of Shi et al. [78] supports public auditing. Similarly, Shacham and Waters [42] proposed a theme to support public auditing based on BLS signatures. They used erasure code encoding, but they did not consider other criteria.

One challenge of public auditing of POR was that the computational burden on the public authentication tag block that users computed with large numbers of files. To address this issue, Li et al. [47] proposed a new cloud storage solution with two independent cloud servers namely cloud storage server and cloud auditing server, which were assumed to be semi-honest. In particular, the tasks they considered allowes the cloud auditing server to represent users to preprocess data and then audits data integrity before uploading the data to the cloud storage server. The cloud eliminated the user's participation in the auditing process and in the pre-processing phase. Public auditing and computation efficiency were offered in this work.

Armknecht et al. [52] introduced the concept of searchability (OPOR) for outsourced certification, in which users could delegate task to external auditors and cloud providers to execute and validate PORs. They argued that OPOR settings have security risks that were covered by the POR security model. To solve this problem, they proposed a formal framework and a security model for OPOR and presented an instantiation of OPOR based on a provably secure private POR scheme provided by Shacham and Waters [42] and evaluated its performance in real cloud settings. Their evaluation results showed the scheme could minimize the user's workload and bring negligible overhead to the auditor compared with the solution of Shacham and Waters [42]. Other criteria except for public auditing and privacy were not mentioned in their article.

In the public auditing POR scheme by Yuan et al. [50], the communication overhead between the prover and the verifier was a constant number of group elements. Unlike existing private POR structures, their approach allowed public auditing and free data owners from being online. They achieved these goals by tailoring and uniquely combining techniques such as constant size polynomial combinations and homomorphic linear certifiers. Their analysis showed that the proposed solution was effective and practical. The only mentioned computation efficiency and public auditing.

After that, Yuan et al. [51] proposed a novel data integrity auditing scheme for multiple-user modification, collusion resistance and constant calculation costs. Their scheme supported public auditing and effective user revocation and proved to be safe. Numerical analysis and extensive experimental results demonstrated the efficiency and scalability of their approach. The two criteria involved were public auditing and security in their article.

We provide a comparision of the POR scheme mentioned above in Table 4.

4.3 Auditing Schemes for Multiple Replicas

However, PDP and POR might not be effective in the situation when data get lost or damaged at the servers. One solution is to store multiple replicas of each file, and use other copies if the original copy is corrupted. In the previous section, we classify auditing schemes on single copy

TABLE 4
COMPARISON OF POR SCHEMES AND POR-BASED SCHEMES

	Low storage	Communication and computation efficiency	Dynamic data operation	Unlimitation of verification	Public auditing	Batch auditing	Privacy preservation	Provable security	Retrievability
[42]				Yes					Yes
[43]			No		Yes				Yes
[44]	Yes	No	No					Yes	Yes
[45]	Yes		No		No				Yes
[46]			Yes						Yes
[47]		Yes			Yes				Yes
[48]			Yes					Yes	Yes
[49]		Yes	Yes						Yes
[50]		Yes			Yes				Yes
[51]					Yes			Yes	Yes
[52]					Yes		Yes		Yes
[78]	Yes		Yes		Yes				Yes
[79]	Yes	Yes	Yes						Yes

Yes: supporting; No: not supporting; Blank: not mentioned.

(schemes based on PDP and POR); in this section, we will-review the auditing schemes on multiple replicas, also known as auditing under distributed nodes.

Data integrity auditing under distributed models was firstly introduced by Deswarte et al. [12]. They provided an example of remote integrity checking under distributed servers on the basic of using checksum. The user periodically sends a request to the server to calculate the checksum of the file, and then the server returned the checksum to the user, who verifies the result by comparing it with the locally stored checksums of data. The calculation of checksums needs hash function, but the secure hash function had not been found. Thus, the security of the scheme is yet to be verified. In addition, this traditional protocol is inefficient and vulnerable to various attackers. In order to resist avoid attacks, the verifier also needs to store a table of checksums on file data locally, which required a lot of storage overhead. Other than this, other criteria were not mentioned in their paper.

Ateniese et al. [55] also think about the distributed storage system because their protocol transmits a small and constant amount of data while the network communication was limited. But later the solution was proved to be unsafe during a collusion attack between multiple servers, so their solution is no longer applied into multiple replicas protocols.

Curtmola et al. [56] proposed a multiple replicas PDP scheme using cryptographic tools to ensure data security and storing multiple different copies of a file on multiple servers in a distributed system. When a file on one server is damaged, the other copies can serve as bakcups. Storing multiple copies was computationally more efficient than storing a single block. The solution might cause additional storage and communication overhead at the client side. It did not support public auditing either. Except for these, they did not cover other criteria.

Based on the work MR-PDP of Curtmola et al. [56], Hao et al. [80] proposed a protocol that supports public verifiability. They used homomorphic authentication tags based on BLS signature to construct the scheme. Security analysis and performance analysis showed that the propotol is secure and efficient.

Zhu et al. [58] considered integrity auditing for data storage in a distributed hybrid cloud. They used homomorphic verification response (HVR) and hash indexing (HIH) techniques to combine responses from multiple servers into one final result and return them to the users. The experiment validated the effectiveness of their solution. Their solutions supported privacy protection and dynamic data manipulation. However, due to homomorphism, users perhaps had no knowledge about which server their file was located on when some errors occurred. Their scheme required a small constant amount of communication overhead, but the storage overhead and other criteria were not involved. Then they highlighted in the paper [59] that the PDP scheme in a collaborative cloud environment should be usable and secure, and that communication and computational overhead should be smaller than non-cooperative. In the paper [59], they used bilinear mapping operations to construct scheme, which supported public auditing, but suffered from high complexity for larger files.

He et al. [60] provided a novel and efficient data possession checking (DMRDPC) scheme for distributed multiple replicas to reduce the communication overhead between servers in different distributed locations. This was achieved by improving the efficiency of scheduling multiple replicas of data to have a check order. The main criterion they focused was the computational overhead.

The increase of replicas and servers in a distributed environment might cause higher complexity. Additionally, users might have transparency issues about the location of the data. To address this issue, Etemad et al. [62] proposed a transparent distributed multiple replicas dynamic PDP scheme whose complexity does not depend on the number of file replicas and servers. They considered the multiple-users scenario for provable version control systems for the first time. In their article, they only talked about computation efficiency.

Considering the public key infrastructure, PDP protocol needs public key certificate distribution and management which brought considerable overhead. Identity-based public key cryptography could eliminate the complicated certificate management and increase the efficiency. To provide identity-based authentication in distributed systems, Wang [63] proposed a model for data integrity auditing: identity-based distributed provable data possession scheme (ID-DPDP). It was a security model, based on bilinear pairing, and provably safe under the assumption of Diffie-Hellman. This paper only discusses computation efficiency and security, but not the other criteria.

Long et al. [65] implemented a PDP scheme of multiple replicas of data storage in the cloud based on the complete node of the AVL tree in their article. By balancing AVL trees, the scheme could support the dynamic operations on data in the cloud, improve search efficiency and verify the location of the data. This model was shown to provide security, reduced communication and storage overhead based on the scheme of Merkle Hash tree. This paper mainly focuses on evaluating computation efficiency and supports dynamic operation.

4.3.1 Error Tracking in Multiple Replicas Auditing

When an error occurs in a file, it is important to know in which replica the errors occur. A few papers have addressed this issue.

Wang et al. [57] proposed a scheme that could locate data error by using distributed homomorphic tokens to perform distributed verification on erasure-encoded data thus could find out which server was in error. Unlike most of the previous work, their solutions also supported safe and efficient dynamic data operations such as data modification, deletions, and appends. Extensive security and performance analysis showed their solution was efficient and could withstand the problem of Byzantine failures, malicious data modification attacks, and even server collusion attacks. However, they did not mention storage and computation overhead, nor did the number of verifications, batch auditing and privacy preservation.

Barsoum et al. [61] further addressed the error-location issue by proposing a pairing-based multiple provable data possession (PB-PMDP) scheme that guaranteed all outsourced data replicas stored in the cloud server and not changed. Moreover, the scheme allowed authorized third parties to access the user's files, in other words, it supported public auditing and proved resistant to server collusion attacks. The experimental results showed that the number of auditing times and the number of replicas of the file were almost independent. In addition, they improved the scheme that could identify damaged copies. Except for public auditing and security proof, other criteria were not mentioned in this article.

Rakesh et al. [64] designed an adaptive and efficient scheme that guaranteed the correctness of user data stored in the cloud with additional features. They used homomorphic tokens to erase distributed authentication of encoded data and identify servers with behavioral errors. Their scheme also supported efficient and secure dynamic data operations, e.g., insertion, deletion and modification. Other criteria we proposed were not discussed in their article.

We give a comparison of schemes in distributed system for multiple replicas in Table 5. In this part of reviewed work, retrivability was not supported if error tracking cannot be supported.

5 OPEN ISSUES AND FUTURE DIRECTIONS

Nowadays, the cloud computing environment is becoming increasingly comprehensive and complex from a public cloud to a private cloud, a single cloud to multiple clouds, a center cloud to a distributed system. Under this circumstance, the reliability of the cloud computing is essential. Furthermore, the cloud computing has broad application prospects, but this can be achieved only when security is guaranteed, which makes our work on computing security timely and important. This paper reviews the basic aspects of cloud computing security and researches various schemes and technologies about integrity auditing for data storage, compares existing schemes according to the given evaluation criteria, pointing out the shortcomings and areas of improvement. By enumerating the existing auditing schemes for ensuring data integrity from a cloud data center to distributed systems, we propose the future research directions. Under the cloud environment, especially as for the distributed system, it is important to consider the communication overhead between users and servers, and the possibilities of reducing user burden by introducing third parties while balancing security. Furthermore, it is also important to enhance the data processing efficiency and data update operation in auditing scheme design.

Based on our review on the above schemes, we discuss the future directions in three aspects and application scenarios.

5.1 Research on Schemes Combining Integrity Auditing and Data Deduplication

In the integrity auditing schemes under the scenario of multiple replicas, after generating multiple copies of a same file, the problem of duplicated storage is unavoidable [66]. It is also important to solve the efficiency problem in order to improve the schemes of integrity auditing [67].

Recently, there are also a few schemes which combines auditing and deduplication. In the paper of Li et al. [68],

TABLE 5
COMPARISON OF AUDITING SCHEMES FOR MULTIPLE REPLICAS

	Low storage	Communication and computation efficiency	Dynamic data operation	Unlimitation of verification	Public auditing	Batch auditing	Privacy preservation	Provable security	Error Tracking
[12]	No	No						No	
[14]		Yes						No	
[56]		No			No				
[57]			Yes		No			Yes	Yes
[58]		No	Yes		Yes		Yes		
[60]		Yes							
[61]					Yes			Yes	Yes
[62]		Yes							
[63]		Yes						Yes	
[64]		Yes	Yes						Yes
[68]		Yes	Yes						
[80]		Yes			Yes			Yes	
[81]	Yes	Yes	No		Yes	Yes			

Yes: supporting; No: not supporting; Blank: not mentioned.

they used the techniques of bilinear pairings and bloom filter to build a public auditing scheme that supports data deduplication. In their scheme, user divided the message into n blocks, and generated a Bloom Filter utilizing a pseudorandom function. They used a constant-size tag generation algorithm. The Bloom filter was used to perform POW (Proof of Work) protocol during deduplication. Their scheme was proven to be uncheatable and anonymous under the BDH assumption in a random oracle model. In the work of Daniel et al. [69], they also proposed a scheme of deduplication and auditing which was lightweight. They combined hashing and symmetric encryption with improved distributed hash table data structure to reduce the communication and computation overhead. The techniques of convergent encryption and filters were used to decrease the storage cost. In the scheme of Yuan et al. [70], they also had designed storage deduplication by using polynomial-based authentication tags and homomorphic linear authenticators.

The research on integrity auditing in cloud computing tends to reach a plateau. However, how to improve the integrity auditing scheme, without the storage efficiency problem under the environment of multiple replicas, ensuring data security in the cloud still seems to be an important research direction in the future.

5.2 Research on Auditing Schemes on Blockchain

The concept of blockchain was first proposed in 2008. The core technologies of blockchain - distributed accounting, asymmetric encryption, consensus mechanism and intelligent contract technology - play an important role in protecting data privacy and security. Its decentralization feature could be widely used in the multiple replicas distributed environment.

With the rise of the blockchain technology, researchers have proposed the scheme of integrity auditing based on it. Li et al. [71] proposed a security framework for cloud data auditing using blockchain technology. Users0̂ operation information on the file is created into a block after being verified by all nodes in the blockchain network, and then appended in the blockchain. Any modification to the data can be checked by a block structure, which ensures the security of the audit data source. They also have used the Merkle hash tree to design their scheme and addressed the secure problem, not only in the servers and the third parties, but also at the users end.

By applying blockchain technology, Xue et al. [72] proposed an identity-based public storage auditing (IBPA) scheme for a cloud storage system, which emphasized the importance of public auditing. This scheme was designed based on bilinear maps and computational Diffie-Hellman assumption. In the IBPA scheme, the nonce of the blockchain was used to construct unpredictable and easily verifiable challenge information, preventing malicious third-party auditors from falsifying auditing results to deceive users. Users only need to verify the TPA's batch auditing results to ensure the integrity of the data stored in the cloud. This scheme was proved to protect data integrity from attacks with efficiency and feasibility.

Along the same line, Qi et al. [73] introduced a framework of Distributed Integrity and Reputation Auditing (DIRA) schemes by using blockchain. The scheme was able to outsource data updates and integrity verification without any primary nodes in a distributed system. Their scheme can detect malicious auditors, and an improved version of blockchain with two-step transaction validation, resistant to history modification.

In the context of blockchain technology, how to implement auditing under distributed multiple replicas storage could also be a promising research direction in the future.

5.3 Research on Integrity Auditing in Edge Computing

Edge computing has natural distribution characteristics and is also an extension of distributed computing. Edge computing, as the name implies, is the computing occurs at the edge, and the characteristics of the edge environment are caused by high latency, low speed, and unreliable network between all sites. The centralized resource pool calculation method in the cloud cannot meet specific services. Delivery and application functions, thus transfer some or all of the processing to end users or data collection points in order to reduce the latency of data calculation and transmission between applications. According to the Openstack White Paper on Edge Computing, edge computing provides cloud computing for application developers and service providers, as well as IT environments at the edge of the network. With the goal of bringing computing, storage, and network resources closer to data producers or end users, edge computing has following characteristics: 1) low latency - the delay experienced by end users is lower than that of relying on central computing and 2) reduced bandwidth requirement and transfer workload for end users. However, edge computing requires a a standardized and unified environment, automation management, and response plans in the event of hardware failure.

As mentioned in [84], data integrity auditing is one of the series of security issues in edge computing [85]. Since edge computing is developed based on cloud computing, auditing work might be designed based on the scheme in cloud computing, while adapting to the characteristics of edge computing mentioned above. For example, since data update frequency is higher, supporting data update operation may be the first consideration when designing auditing schemes. The auditing work might also need to consider data security and privacy protection. Because of the widely distributed servers, the auditing in edge computing needs to interact with multiple server nodes. Compared with traditional auditing schemes, the communication consumption between users and servers may need special consideration, but the burden of users can also be reduced using trusted third parties.

In summary, in traditional cloud computing, the auditing of the data stored in the cloud needs to be fully considered to cope with changes in complex environments from a single cloud to multiple clouds. Since multiple replicas stored in multiple clouds may lead to the problem of duplicate storage, it is important to consider deduplication integrated with data integrity auditing. Furtheremore, it is promising to study auditing in the context of edge computing and by applying blockchain.

6 CONCLUSIONS

In this survey, we discussed the issues of data integrity auditing in cloud computing, surveyed and compared various kinds of auditing schemes by employing ten evaluation criteria. We presentd our evaluation in two parts. One is about the auditing of the single copy of a file in a single cloud, which includs PDP-based and POR-based schemes. The other is the auditing of multiple replicas in multiple clouds in a distributed cloud system. Through an overview of the existing literature about auditing, we found some open issues about the exiting schemes and finally propose the future research directions and applications.

ACKNOWLEDGMENT

This work is sponsored by the NSFC (grants 61672410 and 61802293), the Academy of Finland (grants 308087, 314203 and 335262), the Shaanxi innovation team project (grant 2018TD-007), the National Postdoctoral Program for Innovative Talents (grant BX20180238), the Project funded by China Postdoctoral Science Foundation (grant 2018M633461), and the 111 project (grants B16037).

REFERENCES

[1] Z. Yan, X. Yu, and W. Ding, ÒContext-Aware Verifiable Cloud Computing,Ó *IEEE Access*, pp. 1-1, Feb. 2003.

[2] X. Yu, Z. Yan, and A. Vasilakos, ÒA Survey of Verifiable Computation,Ó Mobile Networks and Applications, May 2017.

[3] X. Yu, Z. Yan, and R. Zhang, “Verifiable Outsourced Computation over Encrypted Data,Ó Information Sciences, Dec 2018.

[4] L. Chen and X. Li, ÒResearch on Provable Data Possession and Recovery Technology in Cloud Storage,Ó Journal of Computer Research and Development, 19-25, 2012.

[5] Z. Qin, S. Wu and H. Xiong, ÒA Review on Data Integrity Auditing Protocols for Data Storage in Cloud Computing,Ó Netinfo Security, 2014, (7), 1-6

[6] W. F. Hsien, C. C. Yang and M. S. Hwang, ÒA Survey of Public Auditing for Secure Data Storage in Cloud Computing,Ó Network Security, 2016, 18(1): 133-142.

[7] C. Tan, M. H. A. Hijazi, Y. Lim and A. Gani, ÒA survey on Proof of Retrievability for cloud data integrity and availability: Cloud storage state-of-the-art, issues, solutions and future trends,Ó Journal of Network and Computer Applications, Mar. 2018.

[8] M. Sookhak, H. Talebian, E. Ahmed and A. Gani, ÒA Review on Remote Data Auditing in Single Cloud Server: Taxonomy and Open Issues,Ó Journal of Network and Computer Applications, Aug. 2014.

[9] M.A. Alzain, B. Soh and E. Pardede, ÒA Survey on Data Security Issues in Cloud Computing: From Single to Multi-clouds,Ó 2013, 8(5): 1068-1078.

[10] W. Wang, X. Du and N. Wang, ÒReview on Security Audit Technology for Cloud Computing,Ó Computer Science, 2017, 44(7), 16-20, 2017.

[11] W. Ding, Z. Yan and R. Deng, ÒPrivacy-Preserving Data Processing with Flexible Access Control,Ó *IEEE Transactions on Dependable and Secure Computing,* pp. 1-1, Dec. 2017.

[12] Y. Deswarte, J.J. Quisquater, and A. Sa•dane, ÒRemote Integrity Checking,Ó *Working Conference on Integrity and Internal Control in Information Systems* (IICIS Ô03), pp. 1-11, 2003.

[13] D.L.G Filho and P.S.L.M Barreto, ÒDemonstrating Data Possession and Uncheatable Data Transfer,Ó *IACR Cryptology ePrint Archive*, pp. 150-164, Jan. 2006.

[14] G. Aiuseppe, R. Burns, R. Curtmola, J. Herring. L, Kissner. Z. Peterson and D. Song. ÒProvable Data Possession at Untrusted Stores,Ó *Proceedings of the 14th ACM conference on Computer and communications security* (CCS Ô07), 598-609, Jan. 2007.

[15] W. Ding, Z. Yan and R. H. Deng, ÒEncrypted Data Processing with Homomorphic Re-Encryption,Ó Information Sciences, 35-55, Oct. 2017.

[16] F. Sebe, J. Domingo-Ferrer, A. M. Balleste and Y. Deswarte, ÒEfficient Remote Data Possession Checking in Critical Information Infrastructures,Ó *IEEE Transactions on Knowledge and Data Engineering,* Vol. 20, no.8, pp. 1034 Ð 1038, Jun. 2008, doi: 10.1109/TKDE.2007.190647.

[17] G. Ateniese, R. D. Pietro, L. V. Mancini and G. Tsudik, ÒScalable and Efficient Provable Data Possession,Ó *Proceedings of the 4th international conference on Security and privacy in communication netowrks* (SecureComm Ô08), Sep. 2008.

[18] G. Ateniese, S. Kamara and J. Katz, ÒProofs of Storage from Homomorphic Identification Protocols,Ó *International Conference on the Theory and Application of Cryptology and Information Security* (ASIACRYPT Ô09), pp. 319-333, Dec. 2009.

[19] C. C. Erway. A. KŸp•Ÿ, C. Papamanthou and R. Tamassia, ÒDynamic Provable Data Possession,Ó ACM Transactions on Information and System Security (TISSEC), vol. 17, no. 15, pp. 213-222, Jan. 2009.

[20] M. A. Shah, M. Baker, J. C. Mogul and R. Swaminathan, ÒAuditing to Keep Online Storage Services Honest,Ó *Proceedings of the 11th USENIX workshop on Hot topics in operating systems* (HOTOS'07), no. 11, May. 2007.

[21] M. A. Shah, R. Swaminathan and M. Baker, ÒPrivacy-Preserving Audit and Extraction of Digital Contents,Ó *IACR Cryptology ePrint Archive*, Apr. 2008.

[22] C. Wang, Q. Wang, K. Ren and W. Lou, ÒPrivacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,Ó *Proceedings of the 29th conference on Information communications* (INFOCOM'10), pp. 525-533, Mar. 2010.

[23] C. Wang, S. S. M. Chow, Q. Wang, K. Ren and W. Lou, ÒPrivacy-Preserving Public Auditing for Secure Cloud Storage,Ó *IEEE Transactions on Computers archive,* vol. 62, no. 2, Feb. 2013.

[24] J. Yang, H. Wang, J. Wang, C. T and D. Y, ÒProvable Data Possession of Resource-constrained Mobile Devices in Cloud Computing,Ó *Journal of Networks,* vol. 6, no. 7, pp. 1033-1040. Jul. 2011.

[25] C. Gritti, W. Susilo and T. Plantard, ÒEfficient Dynamic Provable Data Possession with Public Verifiability and Data Privacy,Ó *Australasian Conference on Information Security and Privacy* (ACISP Ô15), pp. 395-412, Jun. 2015.

[26] H. Wang, ÒProxy Provable Data Possession in Public Clouds,Ó *IEEE Transactions on Services Computing,* vol. 6, no. 4, pp. 551-559, Oct/Dec. 2013.

[27] H. Wang, Q. Wu, B. Qin and J. Domingo-Ferrer, "Identity-based remote data possession checking in public clouds," *Iet Information Security*, 2014.

[28] H. Wang, D. H and S. T, ÒIdentity-Based Proxy-Oriented Data Uploading and Remote Data Integrity Checking in Public Cloud,Ó *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1165-1176, Jun. 2016.

[29] B. Wang, B. Li and H. Li, ÒPanda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud,Ó *Online International Conference on Green Engineering and Technologies* (IC-GET Ô16), May. 2017.

[30] T. Yang, B. Y, H. Wang, J. Li and Z. Lv, ÒCryptanalysis and Improvement of Panda Ð Public Auditing for Shared Data in Cloud and Internet of Things,Ó *Multimedia Tools and Applications*, vol. 76, no. 19, pp. 19411-19428. Dec. 2015.

[31] J. Yuan and S. Yu, ÒPublic Integrity Auditing for Dynamic Data Sharing with Multiuser Modification,Ó *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1717-1726, Aug. 2015.

[32] J. Zhang and Q. Dong, ÒEfficient ID-based Public Auditing for the Outsourced Data in Cloud Storage,Ó *Information Sciences*, vol. 343-344, pp. 1-14, May. 2016.

[33] D. He, H. Wang, J. Zhang and L. Wang, "Insecurity of an identity-based public auditing protocol for the outsourced data in cloud storage," *Information Sciences*, 2017.

[34] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni and K. R. Choo, ÒFuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems,Ó *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 72-83, Feb. 2017.

[35] C. Liu, J. Chen, L. T. Yang, X. Zhang, C. Yang, R. Ranjan and R. Kotagiri, ÒAuthorized Public Auditing of Dynamic Big Data Storage on Cloud with Efficient Verifiable Fine-Grained Updates,Ó *IEEE Transactions on Parallel and Distributed Systems*, vol.25, no. 9, pp. 2234-2244, Sep. 2014.

[36] T. Jiang, X. Chen and J. Ma, ÒPublic Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation,Ó *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2363-2373, Aug. 2016.

[37] J. Shen, J. Shen, X. Chen, X. Huang and W. Susilo, ÒAn Efficient Public Auditing Protocol with Novel Dynamic Structure for Cloud Data,Ó *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402-2415, Oct. 2017.

[38] R. R. Kalangi, M. V. P. C. S. Sekhare, ÒA Novel Multi-user Fingerprint Minutiae Based Encryption and Integrity Verification for Cloud Data,Ó *International Journal of Advanced Computer Research*, vol. 8, no. 37, pp. 161-170, Jul. 2018.

[39] S.K. Nayak and S. Tripathy, ÒSEPDP: Secure and Efficient Privacy Preserving Provable Data Possession in Cloud Storage,Ó *IEEE Transactions on Services Computing*, pp. 1-1, Mar. 2018.

[40] J. Yu and R. Hao, ÒComments on "SEPDP: Secure and Efficient Privacy Preserving Provable Data Possession in Cloud Storage",Ó *IEEE Transactions on Services Computing*, pp. 1-1, Apr. 2019.

[41] A. Li, S. Tan and J. Yan, ÒA Method for Achieving Provable Data Integrity in Cloud Computing,Ó *The Journal of Supercomputing*, vol. 75, no. 1, pp. 92-108, Jan. 2019.

[42] A. Juesl, S. Burton and J. Kaliski, ÒPors: Proofs of Retrievability for Large Files,Ó *Proceedings of the 14th ACM conference on Computer and communications security* (CCS Ô07), pp. 584-597, Jan. 2007.

[43] H. Shacham and B. Waters, ÒCompact Proofs of Retrievability,Ó *International Conference on the Theory and Application of Cryptology and Information Security* (ASIACRYPT Ô08), pp. 90-107, 2008.

[44] Y. Dodis, S. Vadhan and D. Wichs, ÒProofs of Retrievability via Hardness Amplification,Ó *Theory of Cryptography Conference* (TCC 2009), pp. 109-127, Mar. 2009.

[45] K. D. Bowers, A. Juesl and A. Oprea, ÒProofs of Retrievability: Theory and Implementation,Ó *Proceedings of the first ACM Cloud Computing Security Workshop* (CCSW Ô09), Nov. 2009.

[46] Q. Zheng and S. Xu, ÒFair and Dynamic Proofs of Retrievability,Ó *Proceedings of the first ACM conference on Data and application security and privacy* (CODASPY Ô11), pp. 237-248, Feb. 2011.

[47] J. Li, X. Tan, X. Chen and D. S. Wong, ÒAn Efficient Proof of Retrievability with Public Auditing in Cloud Computing,Ó *Proceedings of the 2013 5th International Conference on Intelligent Networking and Collaborative Systems* (INCOS Ô13), pp. 93-98, Sep. 2013.

[48] J. Li, X. Tan, X. Chen, D. S. Wong and F. Xhafa, ÒOPoR: Enabling Proof of Retrievability in Cloud Computing with Resource-Constrained Devices,Ó *IEEE Transactions on Cloud Computing*, vol. 3, no. 2, pp. 195-205, Apr / Jun. 2015.

[49] D. Cash, A. KŸp•Ÿ and D. Wichs, ÒDynamic Proofs of Retrievability via Oblivious RAM,Ó *Journal of Cryptology*, vol. 30, no. 1, pp. 22-57, Sep. 2015.

[50] J. Yuan and S. Yu, ÒProofs of Retrievability with Public Verifiability and Constant Communication Cost in Cloud,Ó *Proceedings of the 2013 international workshop on Security in cloud computing* (Cloud Computing Ô13), pp. 19-26. May. 2013.

[51] J. Yuan and S. Yu, ÒEfficient Public Integrity Checking for Cloud Data Sharing with Multi-user Modification,Ó *IEEE Conference on Computer Communications* (IEEE INFOCOM Ô14), May. 2014.

[52] F. Armknecht, J. Bohli, G. O. Karame, Z. Liu and C. A. Reuter, ÒOutsourced Proofs of Retrievability,Ó *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (CCS Ô14), pp. 831-843, Nov. 2014.

[53] F. Armknecht, J. Bohli, D. Froelicher and G. Karame, ÒSharing Proofs of Retrievability across Tenants,Ó *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (ASIA CCS Ô17), pp. 275-287, Apr. 2017.

[54] N. S. Chavan and D. Sharma, ÒSecure Proof of Retrievability System in Cloud for Data Integrity,Ó *2018 Fourth International Conference on Computing Communication Control and Automation* (ICCUBEA Ô18), Aug. 2018.

[55] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson and D. Song, ÒRemote Data Checking using Provable Data Possession,Ó *ACM Transactions on Information and System Security*, vol. 14, no. 12, May. 2011.

[56] R. Curtmola, O. Khan, R. Burns and G. Atenises, ÒMR-PDP: Multiple-Replica Provable Data Possession,Ó *2008 The 28th International Conference on Distributed Computing Systems* (ICDCS Ô08), Jun. 2008.

[57] C. Wang, Q. Wang, K. Ren and W. Lou, ÒEnsuring data storage security in Cloud Computing,Ó *2009 17th International Workshop on Quality of Service* (IWQoS Ô09), Jul. 2009.

[58] Y. Zhu, H. Wang, Z. Hu, G. Ahn, H. Hu and S. S. Yau, ÒEfficient Provable Data Possession for Hybrid Clouds,Ó *Proceedings of the 17th ACM conference on Computer and communications security* (CCS Ô10), pp. 756-758, Oct. 2010.

[59] Y. Zhu, H. Hu, G. Ahn and M. Yu, ÒCooperative Provable Data Possession for Integrity Verification in Multicloud Storage,Ó *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2231-2244, Feb. 2012.

[60] J. He, Y. Zhang, G. Huang, Y. Shi and J. Cao,Ó Distributed Data Possession Checking for Securing Multiple Replicas in Geographically-dispersed Clouds,Ó *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1345-1358, Sep. 2012.

[61] A. F. Basoum and M. A. Hasan, ÒIntegrity Verification of Multiple Data Copies over Untrusted Cloud Servers,Ó *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (CCGRID Õ12), pp. 829-834, May. 2012.

[62] M. Etemad and A. KŸp•Ÿ, ÒTransparent, Distributed, and Replicated Dynamic Provable Data Possession,Ó *International Conference on Applied Cryptography and Network Security* (ACNS Õ13), pp. 1-18, Jun. 3013.

[63] H. Wang, ÒIdentity-Based Distributed Provable Data Possession in Multicloud Storage,Ó *IEEE Transactions on Services Computing,* vol. 8, no. 2, pp. 328-340, Mar / Apr. 2015.

[64] B. Rakesh, K. Lalitha, M. Ismail and H. P. Sultana, ÒDistributed Scheme to Authenticate Data Storage Security in Cloud Computing,Ó *International Journal of Computer Science & Information Technolog,* vol. 9, no. 6, Dec. 2017.

[65] M. Long, Y. Li and F. Peng, ÒDynamic Provable Data Possession of Multiple Copies in Cloud Storage Based on Full-Node of AVL Tree,Ó *International Journal of Digital Crime and Forensics,* vol. 11, no. 1, Jan / Mar. 2019.

[66] Z. Yan, L. Zhang, W. Ding and Q. Zheng, ÒHeterogeneous Data Storage Management with Deduplication in Cloud Computing,Ó *IEEE Transactions on Big Data,* vol. 5, no. 3, pp. 393-407, Sep. 2019.

[67] Z. Yan, W. Ding, X. Yu, H. Zhu and R. H. Deng, ÒDeduplication on Encrypted Big Data in Cloud,Ó *IEEE Transactions on Big Data,* vol. 2, no. 2, pp. 138-150, Jun. 2016.

[68] C. Li and Z. Liu, ÒA Secure Privacy-Preserving Cloud Auditing Scheme with Data Deduplication,Ó *International Journal of Network Security,* vol. 21, no. 2, pp. 199-210, Mar. 2019.

[69] E. Daniel and N. A. Vasanthi, ÒLDAP: A Lightweight Deduplication and Auditing Protocol for Secure Data Storage in Cloud Environment,Ó *Cluster Computing,* vol. 22, no. 1, pp. 1247-1258, Jan. 2019.

[70] J. Yuan and S. Yu, ÒSecure and Constant Cost Public Cloud Storage Auditing with Deduplication,Ó *2013 IEEE Conference on Communications and Network Security* (CNS Õ13), Dec. 2013.

[71] C. Li, J. Hu, K. Zhou, Y. Wang and H. Deng, ÒUsing Blockchain for Data Auditing in Cloud Storage,Ó *International Conference on Cloud Computing and Security* (ICCCS Õ18), PP. 335-345. Sep. 2018.

[72] J. Xue, C. Xu, J. Zhao and J. Ma, ÒIdentity-based Public Auditing for Cloud Storage Systems Againist Malicious Auditors via Blockchain,Ó *Science China Information Sciences,* Mar. 2019.

[73] Y. Qi and Y. Huang, ÒDIRA: Enabling Decentralized Data Integrity and Reputation Audit via Blockchain,Ó *Science China Technological Sciences,* vol. 62, no. 4, pp. 698-701, Apr. 2019.

[74] A. Barosum, "Provable Data Possession in Single Cloud Server: A Survey, Classification and Comparative Study," *International Journal of Computer Applications*, 2015.

[75] M. Thangavel, P. Varalakshmi, R. Sindhuja, "A survey on provable data possession in cloud storage," *Eighth International Conference on Advanced Computing*, 2017.

[76] Y. Dong, L Sun, D. Liu, M. Feng and T. Miao, "A Survey on Data Integrity Checking in Cloud," *International Cognitive Cities Conference*, 2018.

[77] L. Zhou, A. Fu, S. Yu, M. Su and B. Kuang, "Data integrity verification of the outsourced big data in the cloud environment: A survey," *Journal of Network & Computer Applications*, 2018.

[78] E. Shi, E. Stefanov and C. Papamanthou, "Practical dynamic proofs of retrievability," *Computer and Communications Security*, 2013.

[79] Z. Mo, Y. Zhou and S. Chen, "A Dynamic Proof of Retrievability(POR) Scheme with O(kogn) Complexity," *International Conference on Communications,* 2012.

[80] Z. Hao and N. Yu, "A Multiple-Replica Remote Data Possession Checking Protocol with Public Verifiability," *International Symposium on Data, Privacy, and E-commerce*, 2010.

[81] J. Li, H. Yan and Y. Zhang, "Efficient Identity-based Provable Multi-Copy Data Possession in Multi-Cloud Storage," *IEEE Transactions on Cloud Computing*, 2019.

[82] P. Hunt, M. Konar, F.P. Junqueira and B. Reed, "ZooKeeper Wait-free coordination for Internet-scale systems," *Usenix Annual Technical Conference.* 2010.

[83] A. Bessani, M. Correia, B. Quaresma, F. Andre and P. Sousa, "DEPSKY: Dependable and secure storage in a cloud-of-clouds," *ACM Transactions on Storage*, 2011.

[84] D. Liu, Z. Yan, W. Ding and W. Atiquzszaman, "A Survey on Secure Data Analytics in Edge Computing," *IEEE Internet of Things Journal*, 2019.

[85] B. Cao, L. Zhang, Y. Li, D. Feng and W. Cao, "Intelligent Offloading in Multi-Access Edge Computing: A State-of-the Art Review and Framework," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 56-62, March 2019.

[86] Y. Zhang and M. Blanton, "Efficient dynamic provable possession of remote data via balanced update trees," *computer and communications security*, 2013.

[87] G. Yao, Y. Li Y, L. Lei L, H. Wang and C. Lin. "An Efficient Dynamic Provable Data Possession Scheme in Cloud Storage," *grid and pervasive computing*, 2016.

Angtai Li is currently working toward the MasterÕs degree in network security from the School of Cyber Engineering, Xidian University.

Yu Chen is an assistant professor in the School of Information Systems and Technology in the Lucas College of Business at San Jose State University. Prior to that, she was a postdoctoral researcher at University of California, Irvine. Dr. Chen received her PhD ifrom Swiss Federal Institute of Technology at Lausanne, Master's degrees from both Aalto University and Northwestern University of Science and Technology, and Bachelor's degree from Huazhong University of Science and Technology.

Zheng Yan (MÕ06, SMÕ14) received the BEng degree in electrical engineering and the MEng degree in computer science and engineering from the XiÕan Jiaotong University, XiÕan, China in 1994 and 1997, respectively, the second MEng degree in information security from the National University of Singapore, Singapore in 2000, and the licentiate of science and the doctor of science in technology in electrical engineering from Helsinki University of Technology, Helsinki, Finland. She is currently a professor at the Xidian University, XiÕan, China and a visiting professor at the Aalto University, Espoo, Finland. Her research interests are in trust, security, privacy, and securityrelated data analytics. Prof. Yan serves as a general or program chair for 30+ international conferences and workshops. She is a steering committee co-chair of IEEE Blockchain international conference. She is also an associate editor of many reputable journals, e.g., IEEE Internet of Things Journal, Information Sciences, Information Fusion, JNCA, IEEE Access, SCN, etc.

Xiaokang Zhou (MÕ12) received the Ph.D. degree in human sciences from Waseda University, Japan, in 2014. From 2012 to 2015, he was a research associate with the Department of Human Informatics and Cognitive Sciences, Faculty of Human Sciences, Waseda University, Japan. From 2016, he has been a lecturer with the Faculty of Data Science, Shiga University, Japan. He also works as a visiting researcher in the RIKEN Center for Advanced Intelligence Project (AIP), RIKEN, Japan, from 2017. Dr. Zhou has been engaged in interdisciplinary research works in the fields of computer science and engineering, information systems, and social and human informatics. His recent research interests include ubiquitous computing, big data, machine learning, behavior and cognitive informatics, cyber-physical-social-system, cyber intelligence and cyber-enabled applications. Dr. Zhou is a member of the IEEE CS, and ACM, USA, IPSJ, and JSAI, Japan, and CCF, China.

Shohei Shimizu is a Professor at the Faculty of Data Science, Shiga University, Japan and leads the Causal Inference Team, RIKEN Center for Advanced Intelligence Project. He received a Ph.D. in Engineering (Statistical Science) from Osaka University in 2006. His research interests include statistical methodologies for learning data generating processes such as structural equation modeling and independent component analysis and their application to causal inference. He received Hayashi Chikio Award (Excellence Award) from the Behaviormetric Society in 2016. He is a coordinating editor of Springer Behaviormetrika since 2016.