A Survey on Data Integrity Checking in Cloud

Yihui Dong
School of Computer and Software
Nanjing University of Information Science & Technology
Nanjing, China
dongyihuichn@163.com

Le Sun
School of Computer and Software
Nanjing University of Information Science & Technology
Nanjing, China
864457960@qq.com

Dengzhi Liu
School of Computer and Software
Nanjing University of Information Science & Technology
Nanjing, China
liudzdh@126.com

Meng Feng
School of Computer and Software
Nanjing University of Information Science & Technology
Nanjing, China
fengmeng1031@163.com

Tiantian Miao
School of Computer and Software
Nanjing University of Information Science & Technology
Nanjing, China
18362086690@163.com

***Abstract*—Cloud storage which provides efficiency and convenience has been paid more and more attention by not only users but also researchers. Users can save their local costs by outsourcing the data to the cloud. While enjoying the benefits of cloud storage, the users are also worried about whether their data remain intact in the cloud or whether their private information has been leaked. This makes them hesitate to outsource their data and cannot fully trust the cloud servers. Because of the concern of users, data integrity becomes an essential security factor of cloud storage. Many scholars have dedicated themselves to the research of data integrity checking for users' security requirements. In this paper, we survey the latest protocols about data integrity checking in cloud and analyze some important requirements and issues of this kind of protocols. The merits and demerits of the protocols are discussed, and the performance is compared with each other as well.**

Keywords—Cloud storage, data integrity, auditing, security

I. INTRODUCTION

With the development of information technique, more and more data are generated and need to be dealt with. Due to the finiteness of computing capabilities and storage space of personal computers, it is urgent for people to find a new way to solve the problems arising from a large number of data. In recent years, the emergence of cloud computing satisfies people's needs in this area. Cloud computing is a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet [7]. Distributed computing systems have developed with the progress of technology and the needs of society and have evolved from the earliest data sharing needs to the current serverless architecture. As a new technology developed from distributed computing, cloud computing has attracted the attention of individuals, enterprises and research institutes.

Cloud storage which is an important component of cloud computing brings great convenience to users. It is a emergent scheme that puts the storage resources into the cloud and allows users to store or obtain data. Users will obviate the costs of building and maintaining a private storage infrastructure if they outsource their data in the cloud [8]. They don't have to worry about the lack of computing capabilities and storage space. Besides, they can conveniently access their data anytime and anywhere.

Although it is obvious that cloud storage has a lot of benefits, security issues still exist. Data integrity is one of the most significant security concerns since the data outsourced in cloud servers are not physically possessed by the users and the control of data is not in the hands of them. While some users may be willing to sacrifice their privacy for benefits brought from software services, enterprises and government organizations will never do that [8]. As a primitive, data integrity checking guarantee that users' data keep intact on the cloud servers. On other kind of distributed computing systems, signature, digest, replica or other methods are used to guarantee data integrity. The concrete methods of data integrity checking in cloud have been studied by many scholars for years, resulting in the continuous emergence of new techniques [1-6, 10-14]. In this paper, we discuss the research results of some existing data integrity checking protocols.

The rest of this paper is organized as follows. Section II offers the essential requirements of data integrity checking protocols. Section III analyzes the main research issues in data integrity checking. Section IV sums up some data integrity checking protocols which are proposed recently. Section V compares the features of the protocols. Finally, the conclusions are drawn in Section VI.

II. REQUIREMENTS OF DATA INTEGRITY CHECKING PROTOCOLS

Data integrity checking protocols are required to realize the security of data and make users be at ease about outsourcing their data. The protocols will be much easier to be accepted if they satisfy the following requirements.

A. Storage Correctness

Ideally, cloud service provider (CSP) always perform normal operations. However, in practical situations, CSP may generate a report which indicates that the data are intact for its interests even if partial data are tampered with or lost [9].

978-1-5386-5059-2/18/$31.00 ©2018 IEEE
DOI 10.1109/IC3.2018.00031

Hence, the protocols need to assure users that their data are the same as what were stored before.

B. Public Auditability

In some schemes, users have to verify the data integrity by themselves [10]. This shows that users utilize their own resources to complete the verification tasks. A third party auditor (TPA) can perform data integrity checking on behalf of users to eliminate their verification burden.

C. Privacy-Preserving

While the advantages of the existence of TPA are clear, it may be curious. It may attempt to find out the real content of the outsourced data which makes users' data be in danger. This is one of the situations the users don't want to see at all. A valid data integrity checking protocol has the capability to prevent TPA from obtaining private information during the course of verification.

D. Batch Auditing

It is more common for TPA to receive multiple verification tasks from different users in a short period in practical application. In order to solve the problem of inefficiency caused by auditing separately, these tasks can be handled simultaneously which is called batch auditing. It can improve auditing efficiency and also reduce the cost of auditing process.

E. Data Dynamics

We can simply speculate that the data users need at different times are not always the same. Therefore, the users should have the ability to update their outsourced data such as inserting, deleting and modifying due to various reasons. In the process of data integrity checking, security also requires to be guaranteed for the profit of users.

F. Key-Exposure Resilience

Key exposure is another important security risks for data integrity checking and it gets a lot of attention these years. The occurrence of key exposure can cover up the fact of data loss and convince users that the data are still intact. To avoid this, key-exposure resilience should be considered in a sound data integrity checking protocol.

III. Main Research Issues in Data Integrity Checking

While data integrity checking in cloud is being studied by many scholars, there are some issues which have not been addressed very well. They are described as follows.

A. Data Security

Data security may be the most important issue which users are really concerned with. It is an essential characteristic of a mature data integrity checking protocol. Lots of reasons can lead to security threats such as malicious cloud, curious auditors, key exposure, external adversaries and so on. Only when the security of data is guaranteed will the users be assured of the presence of data in cloud. However, new security issues appear from time to time while old issues have been solved. Hence, there still exists difficulties which need to be conquered.

B. Overhead

Trying to reduce the overhead is an important direction of designing a new data integrity checking protocol. During the updating and verification processes, the protocols would beget computation and communication overhead. If multiple users' data have a requirement for verification continually, the auditors are required to have enormous computation capability and enormous verification delay will be emerged. The less the overhead, the higher the efficiency of data integrity checking.

C. Invalid Files

In the batch auditing schemes, a single invalid file can lead to the failure of the whole batch auditing. Then querying the invalid file will bring heavy computation and communication overhead which has a serious effect on batch auditing. How to solve this issue is of great significance to the practicality of the batch auditing schemes.

D. Group User Revocation

The users in the same group need to share the data outsourced in the cloud. If a user is revoked and leave the group, he/she may attempt to obtain the real content of the data. Besides, the revoked users may collude with the cloud server to make fake data look valid.

IV. Data Integrity Checking Protocols

The basic system model which is widely used in many data integrity checking protocols is shown in Fig. 1. CSP and data owners interact with each other through data transmission. Data owners delegate TPA for data integrity checking and receive the check results. TPA sends challenge message to CSP and verifies the integrity of data through the proof sent by CSP.

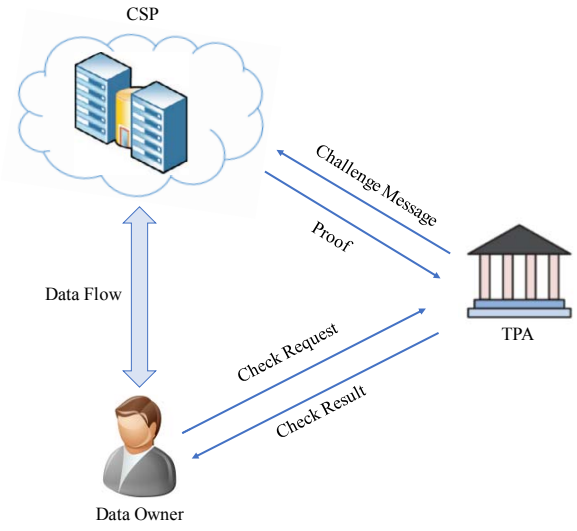

Fig. 1. Basic system model

We divide the selected data integrity checking protocols into two categories, that is, non ID-based encryption protocols and ID-based encryption protocols. And we give discussion on these protocols.

A. Protocols of Non-ID Based Encryption

1) Verification from Indistinguishability Obfuscation: In 2017, Y. Zhang *et al*. [1] proposed a public verification scheme for the cloud storage using indistinguishability obfuscation. Indistinguishability obfuscation is used to ensure the security and reduce the delay and computation overhead on the auditor side which is the aim of the scheme. The auditor doesn't need to have strong computation capability for data integrity checking and only be required to compute a MAC tag. Most computation is delegated to the cloud. The scheme is extended to support not only batch verification but also data dynamic operations which use the Merkle hash tree technique. The auditor can handle multiple tasks from different users simultaneously and the users can update their outsourced data. While the computation overhead of the auditor is linear with the size of the verified data set, it is independent of the size of the data set in this scheme. Nevertheless, it is not efficient for users to generate the obfuscated program and for cloud server to perform the obfuscated program. The scheme also cannot resist malicious auditors.

2) Lightweight Privacy-Preserving: In real life, users' end devices may have low computation capabilities, however, many existing privacy-preserving auditing protocols assume that users' end devices have enough capabilities to compute expensive operations in real time. In 2016, J. Li et al. [2] proposed two lightweight privacy-preserving public auditing protocols to this problem. Because of online/offline signatures, an end device doesn't have to perform heavy computations. In the basic protocol, partial signatures of the whole data are needed to store by the TPA. The TPA will be under great pressure if the data are enormous. Therefore, the basic protocol is only suitable for short data. In the improved protocol, this restriction is eliminated by using the Merkle Hash Tree authentication structure. The scheme also supports batch auditing and data dynamics. Although the scheme can reduce the computation cost on user side, maybe there is a better way to reduce the cost on other side simultaneously.

3) Key-Exposure Resilient Auditing: J. Yu *et al*. [3] proposed a scheme about strong key-exposure resilient auditing for secure cloud storage in 2017. The scheme uses an efficient key update technique, and the key exposure in one time period doesn't have an influence on the security of auditing in other time periods. More specifically, in each time period, the TPA generates an update message and send it to the client. Then the client updates the signing secret key by using the private key and the update message. The malicious cloud cannot obtain the signing secret key when the key is not exposed. Besides, the outsourced data doesn't need to be fixed initially. However, the scheme doesn't support batch auditing or data dynamics which has more space for further research.

B. Protocols of ID-based Encryption

1) ID-Based Data Outsourcing: Y. Wang *et al*. [4] proposed an ID-based data outsourcing scheme to address some outsourcing security issues. The scheme allows users to authorize proxies to upload data to the cloud. Anyone who is unauthorized cannot represent the data owner to upload data. The data owners, proxies and auditors have their own identities, which removes complicated certificate management and embodies the efficiency of the protocol. In the scheme, the auditor can not only verify the data integrity but also audit the information about the origin, type and consistence of outsourced data. The scheme can find any unauthorized behavior about modifying outsourced data and any misuse/abuse of delegations/authorizations which makes it the first scheme to achieve both aims. Users can re-randomize their private keys and the proxies can re-randomize the received delegations. However, both of private keys and delegations should be fixed when handling and outsourcing a file. This is an issue remains to be solved.

2) ID-based Proxy-Oriented Protocol: H. Wang *et al*. [5] proposed a scheme focuses on ID-based proxy-oriented data uploading and remote data integrity checking. The data owner will delegate a proxy to process and upload his data when he is restricted to access the cloud server. The scheme can realize private remote data integrity checking, delegated remote data integrity checking, and public remote data integrity checking based on the original client's authorization. The efficiency is also manifested because of ID-based public key cryptology. Nevertheless, the scheme has a probability of detecting no corruption if the corruption really happens. It will give the attackers a chance to destroy the normal execution of the protocol.

3) Key-Homomorphic Cryptographic Primitive: Y. Yu *et al*. [6] proposed an identity-based remote data integrity checking protocol by using key-homomorphic cryptographic primitive. It can reduce the system complexity and the cost caused by public key authentication framework in other schemes. In the process of verifying, the verifier will not meet information leakage. The scheme first formalizes the security model of zero-knowledge privacy against the TPA in this kind of protocols. Soundness is also proved in the scheme. However, the scheme doesn't have a complete solution to a malicious cloud which may affect the integrity of the data.

V. COMPARISON

To emphasize the differences among the above-mentioned data integrity checking protocols, we compare them in this section. Table I gives the comparisons of functions, including public auditability, batch auditing, data dynamics, and key-exposure resilience.

Then, we evaluate these schemes by comparing the computation overhead in Table II. Here, c denotes the challenged block number, s denotes the sector number of a data block, M and E denote one multiplication and one exponentiation in a cyclic group, respectively, M_q and A_q

TABLE I. COMPARISON OF FUNCTIONS

Protocols	Public auditability	Batch auditing	Data dynamics	key-exposure resilience
Y. Zhang *et al*. [1]	√	√	√	×
J. Li *et al*. [2]	√	√	√	×
J. Yu *et al*. [3]	√	×	×	√
Y. Wang *et al*. [4]	√	×	×	×
H. Wang *et al*. [5]	√	×	×	×
Y. Yu *et al*. [6]	√	×	×	×

TABLE II. COMPARISON OF COMPUTATION OVERHEAD

Protocols	Server	Client
Y. Zhang *et al*. [1]	$(2c+s)E+(3c+s+1)M_q+cA_q+2P+cH$	$2H$
J. Li *et al*. [2]	$(c+1)E+(2c+1)M_q+(c+1)A_q+P+H$	$3E+M+(c+2)M_q+cA_q+2P+cH$
J. Yu *et al*. [3]	$cE+(c-1)M+cM_q+(c-1)A_q$	$(c+2)E+(c+2)M+(c-1)A_q+3P$
Y. Wang *et al*. [4]	$cE+(c-1)M+scM_q+s(c-1)A_q$	$(c+s+2)E+(2l+\ell+s+c+3)M+(c-1)A_q+6P+(c+3)H$
H. Wang *et al*. [5]	$cE+(c-1)M$	$(c+3)E+(c+1)M+2P$
Y. Yu *et al*. [6]	$E+2cM+2P$	$2cE+cP+cH$

respective denote one multiplication and one addition in Z_q, P denotes one bilinear pairing evaluation, H denotes one hash evaluation, l and ℓ are determined by a security parameter.

From the comparison, we can draw some viewpoints. Zhang's [1] and Li's [2] schemes have more functions although they don't have the function of key-exposure resilience. In terms of computation overhead, Zhang's [1] scheme is the least one on the client side. From the general consideration, Wang's [5] and Yu's [6] schemes have the less computation overhead.

VI. CONCLUSION

Data integrity checking is a very meaningful research field in cloud at present. It has a great impetus for the development of cloud storage. The satisfaction of the users is positively related to the degree of perfection of the data integrity checking protocols. For the better protocols, users are more willing to accept and feel relieved of the data in the cloud. In this paper, we analyze several latest protocols and describe their advantages and disadvantages. We also make lists to compare their performance. The practicability of data integrity checking protocols is in the process of improving. However, when the old issues are solved, new issues arise at the same time. There always exist difficulties which threaten the security of outsourced data. In future studies, how to reduce computation and communication cost of data integrity checking protocols can still be researched. Besides, how to achieve better security while ensuring multiple functions are implemented is also a research direction. From this paper, we can summarize some research methods about data integrity which can help us make a contribution in this area.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China under Grant No. 61672295, No. 61672290, No. 61772280 and No. U1405254, the State Key Laboratory of Information Security under Grant No. 2017-MS-10, the 2015 Project of six personnel in Jiangsu Province under Grant No. R2015L06, the CICAEET fund, and the PAPD fund.

REFERENCES

[1] Y. Zhang, C. Xu, X. Liang, *et al*., "Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation," IEEE Transactions on Information Forensics and Security, vol. 12, no. 3, pp. 676–688, 2017.

[2] J. Li, L. Zhang, J. K. Liu, H. Qian, and Z. Dong, "Privacy-preserving public auditing protocol for low-performance end devices in cloud," IEEE Transactions on Information Forensics and Security, vol. 11, no. 11, pp. 2572–2583, 2016.

[3] J. Yu and H. Wang, "Strong key-exposure resilient auditing for secure cloud storage," IEEE Transactions on Information Forensics and Security, vol. 12, no. 8, pp. 1931–1940, 2017.

[4] Y. Wang, Q. Wu, B. Qin, *et al*., "Identity-based data outsourcing with comprehensive auditing in clouds," IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 940–952, 2017.

[5] H. Wang, D. He, and S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," IEEE Transactions on Information Forensics and Security, vol. 11, no. 6, pp. 1165–1176, 2016.

[6] Y. Yu, H. A. A. Man, G. Ateniese, *et al*., "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 767–778, 2017.

[7] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," Grid Computing Environments Workshop, 2008.

[8] S. Kamara and K. Lauter, "Cryptographic cloud storage," International Conference on Financial Cryptography and Data Security, pp. 136–149, 2010.

[9] M. Sookhak, A. Gani, H. Talebian, *et al*., "Remote data auditing in cloud computing environments: A survey, taxonomy, and open issues," ACM Computing Surveys, vol. 47, no. 4, 2015.

[10] A. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," ACM Conference on Computer and Communications Security, pp. 583–597, 2007.

[11] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, and D. X. Song, "Provable Data Possession at Untrusted Stores," ACM Conference on Computer and Communications Security, pp. 598–609, 2007.

[12] 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.

[13] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Transaction on Computers, vol. 62, no. 2, pp. 362–375, 2013.

[14] J. Yuan and S. Yu, "Public Integrity Auditing for Dynamic Data Sharing with Multiuser Modification," IEEE Transactions on Forensics and Security, vol. 10, no. 8, pp. 1717–1726, 2015.