A NEW FRAMEWORK FOR VERIFICATION OF CLOUD DATA STORAGE SYSTEMS

Mr. Bukya Ramachandra Naik[1]
3rd Year Student,
Department of Computer Science,
SV U CM & CS, Tirupati.

Prof. S. Rama Krishna[2],
Professor,
Department of Computer Science,
SV U CM & CS,, Tirupati.

Abstract: Cloud storage services allow users to outsource their data to cloud servers to save local data storage costs. However, unlike using local storage devices, users do not physically own the data stored on cloud servers and are constantly worried about the data integrity of the cloud-stored data. Many public verification schemes have been proposed to allow a third-party auditor to verify the integrity of outsourced data. Most of these schemes are vulnerable in the case that auditors are malicious, and thereby bear a strong assumption: the auditors are honest and reliable. Moreover, in most of these schemes, an external adversary, who is active and online, can modify the outsourced data and tamper with the interaction messages between the cloud server and the auditor, thus invalidating the outsourced data integrity verification. In this article, we propose an efficient and secure public verification of data integrity scheme that resists against external adversaries and malicious auditors. In the proposed scheme, we adopt the random masking technique to fight against the external adversary. We require users to audit auditors' behaviors to resist that malicious auditors fabricate verification results. We also use Bitcoin to construct unbiased challenging messages to thwart the collusion between malicious auditors and cloud servers. Finally, we conduct a performance analysis to demonstrate that the proposed scheme is efficient in terms of the user's auditing overhead.

Index Terms—Cloud storage, data integrity, external adversaries, malicious auditors.

INTRODUCTION

Cloud storage services enable users to outsource their data to cloud servers and access the outsourced data remotely whenever they have Internet connections. Such services provide users an efficient and flexible way to manage their data without deploying and maintaining the local storage device andservice [1], [2], [3], [4]. Specifically, one can process her data on PC, outsource the processed data to cloud servers, and use the data on other devices (e.g. mobile phone) anywhere. Users enjoy great convenience from such services, and it leads to the growing number of cloud storage providers [5], [6].

Despite the benefits brought by the cloud storage service, critical security concerns in data outsourcing have been raised seriously. One of the most important security concerns is data integrity. Because users do not physically own their data once outsourcing the data to cloud servers, they are always worried about the data integrity, i.e. whether their data remains intact on the cloud servers [7]. Concretely, a cloud service provider may hide data loss incidents in order to maintain his reputation [8]. Meanwhile, the cloud service provider may discard the data that has rarely accessed to save the storage space, but claim that no data loss has occurred. Moreover, an external adversary may distort users' data on the cloud servers for financial or political reasons. As such, an efficient and secure verification method is often required by the users to ensure the integrity of their data [9].

Some schemes rely on users themselves to perform the verification, that is, a user has to actively and repeatedly engage in the process with expensive communication and computation overhead, which assumes that a user always has a device with sufficient computation capability and Internet bandwidth to perform the integrity verification. To reduce the verification burden on users, a public verification paradigm has been proposed. The idea is to employ an external and independent auditor to periodically verify the data integrity on behalf of users.

In existing public verification schemes, the auditor is assumed to be honest and cannot be corrupted. But this is a strong assumption as corruption of auditors could happen in practice. In other words, existing public verification schemes are vulnerable in the case that auditors are malicious. Particularly, a malicious auditor can always claim that the outsourced data is (not) retained well in the cloud, no matter what the verification result is, even the malicious auditor would not perform the verification. In addition, the vulnerability of existing schemes is further exacerbated by the fact that the malicious auditor colludes with the cloud server and generates a biased challenging message to check the data blocks which are not corrupted, and thus deceiving the user. Therefore, constructing an efficient public verification of data integrity for cloud storage against malicious auditors is of paramount importance [10].

Some public verification schemes cannot fight against external adversaries, where an active and online adversary can intrude into the cloud server, modify the outsourced data, tamper with the interaction messages between the cloud server and the auditor, and finally pass the auditor's verification. A common approach to resist such adversaries is that the cloud server interacts with the auditor by using a secure channel. However, constructing a secure channel for each verification task requires cumbersome overhead. Therefore, how to resist such adversaries without secure channels is worth to be further studied.

This article investigates how to efficiently verify the integrity of outsourced data against external adversaries and malicious auditors. We propose a novel public verification scheme. In the proposed scheme, we adopt the random masking technique instead of secure channels between cloud servers and auditors to resist the external adversary. In addition, auditors' behaviors should be audited by users to thwart that the malicious auditor invalidates the scheme by fabricating verification results. Furthermore, we use Bit coin to construct the unbiased challenging message, which fights against the malicious auditor's deceiving by colluding with cloud servers. We provide the security analysis to show that the proposed scheme can achieve the security goals, and evaluate the efficiency through simulations.

In the remainder of this article, we first overview the public verification of data integrity technique for cloud storage systems. Then we identify the key influencing factors of the practical public verification scheme and the state-of-the-art research. Based on the design goals identified, we overview a basic public verification scheme and its improved version as an example of the efficient and secure approach toward theory and practice. Next, we evaluate the proposed scheme in terms of security and efficiency. We close the article with concluding remarks.

II METHODOLOGY

BASIC PUBLIC VERIFICATION SCHEME

The influencing factors of systems criteria and crypto criteria listed above in public verification of data integrity are not isolated and can be closely related, which brings a comprehensive evaluation on the public verification technique from cryptography and engineering. In this section we first overview a basic public verification scheme of data integrity proposed by Shacham and Waters [13], which achieves some of the above requirements. Then we show how an external adversary deceives the auditor. Furthermore, such construction assumes that the auditor is honest and cannot be corrupted, we also show how a malicious auditor invalidates the scheme.

A. Warm-up Scheme

We first review the public verification scheme proposed by Shacham and Waters (SWP for short), a user , a cloud server and an auditor are involved in the SWP. The SWP consists of the following algorithms.

Setup. With a security parameter , determines the bilinear map: $e : G \square G \square GT$ [13]. Then chooses secretparameters $\square$, and generates the public parameters1 $(, ,...,) \; s \; v \; g \; u \; u \square \square$, where g is the generator of the multiplicative group G .

Store. transforms his data M into n blocks, and further splits each block into s sectors. chooses a random element *name* for file naming and computes a file tag $\square$ on *name* , which enables to check the consistency of the data file that to be checked. In other words, according to $\square$, can determine the checked data file is the data file should be checked. It ensures that cannot deceive by using other valid data files. Then, generates a tag $i \square$ for each data block. $i \square$ is based on the BLS signature [15], which is the HLA, and has an appealing feature that multiple tags can be aggregated into a short one, where the size of the aggregated tag is independent of the number of the tags to be aggregated, and a verifier can verify the validity of the aggregated tag instead of verifying the tags one by one. Finally, outsources the datafile, file tag, and $i \square$ into .

Audit. For each verification task, first determines $I$and randomly chooses $,i \; v \; i \square I$, where I is a random subsetof set $\{1,...,n\}$ to determine which data blocks should be verified, $i \; v$ is a random element for each verification. Next, sends the challenging message $\{(,)\}( ) \; i \; chal \square \; i \; v \; i \square I$to.

Prove. After receiving *chal*, first verifies $\square$, and then responses i , ([1,]) $vi \; j \; i \; iji \; IiI$

$v \; m \; j$ sto as the corresponding proof information. Observe that for each verification, $i \; v$ is different, it ensures the freshness of the proof information. Due to the appealing feature of the HLA, multiple data blocks and multiple signatures can be aggregated into as hort one (i.e. $j \square$ and $vi \; I \; iI$), which guarantees the minimal communication overhead.

Verify. Upon receiving the proof information, verifies the data integrity by checking

$1(,) ((\|) ,) \; i \; jsvji \; I \; je \; g \; e \; H \; i \; name \; u \; v,$

Where $H(\square)$ is a BLS hash [15].

B. On the Vulnerability Against External Adversaries In the SWP, the proof information generated by is[1,] { , } $j \; j$ Observe that $j \; i \; ij \; i \; Iv \; m$ when an external and active varsity intrudes into the , modifies each data block To deceive and pass the verification, the versary can eavesdrop the challenging message chal , intercept the proof information and compute Finally, the adversary sends the modified proof information to, and the modified data file passes the verification. C. On the Vulnerability Against Malicious Auditors Now, we argue how malicious auditors invalidate the SWP.A corrupted auditor can deceive and in the following ways. First and the most simple way, a malicious auditor can always claim that the outsourced data is (not) retained intact in the cloud, no matter what the verification result is, even the malicious auditor will not perform the verification. Since and trust the auditor, they will accept the auditor's claim without

doubt. Second, the malicious auditor colludes with to deceive. In this case, the outsourced data have been corrupted, but the auditor generates a biased challenging message to check the data blocks which are not corrupted. Third, the malicious auditor colludes with to circumvent. That is, the outsourced data is retained well, but the auditor

claims that the data have been corrupted. The SWP cannot deter against malicious auditors, thus it hasto bear a strong assumption: the auditors are honest and reliable. How to resist against malicious auditors is worth to be further studied.

ENHANCED CONSTRUCTIONS AGAINST MALICIOUSAUDITORS AND EXTERNAL ADVERSARIES

As we discussed before, malicious auditors and external adversaries can invalidate the SWP. Most of existing public verification schemes follow the SWP, and thus have the same framework and threat model. As a consequence, these schemes also cannot fight against external adversaries and malicious auditors. The external adversary can invalidate the SWP, since there exists definite linear relationship between the proof information(i.e. j) and the data blocks (i.e. ij m). To resist external adversaries without secure channels, the random masking technique is adopted in the computation of the proof information. Specifically, we employ random masking as an on linear disturbance code to change the definite linear relationship between the proof information and the data blocks to nonlinear relationship.

To resist malicious auditors, the auditor's behavior should be checked. That is, whether the auditor performs the established verification indeed. Therefore, in the enhanced scheme, the auditor is required to generate an entry for each verification task, and store it into a log file. The user audits the auditor's behavior by checking the validity of the log file. This guarantees that the malicious auditor cannot fabricate a verification result to deceive the user and/or the cloud server. Here, we want to further emphasize that the periodicity the user audits the auditor is much longer than the periodicity the auditor verifies the data integrity.

However, such paradigm cannot deter against malicious auditors perfectly, since the malicious auditor still can deceive the user by generating a biased challenging message, where the corrupted data blocks will never be checked. It is also impractical to require the user to generate a new challenging message for each verification task, due to security and efficiency reasons. To address this problem, we use Bit coin to construct the challenging message. Bit coin has an attractive property. That is, given a determinate time t , if t is a past or current time, we can easily find a Bit coin block, which is generated in the nearest time of t ; but if t is a future time, the Bit coin block, which is generated in t , is unpredictable. Here, we denote the hash of Bit coin block, which is generated in a past time t as t Bl . Since Bit coin has this property, we can consider Bit coin as a time-based pseudo randomness source. The output of this source can be computed when the input of the source is a past/current time, otherwise, the output is unpredictable.

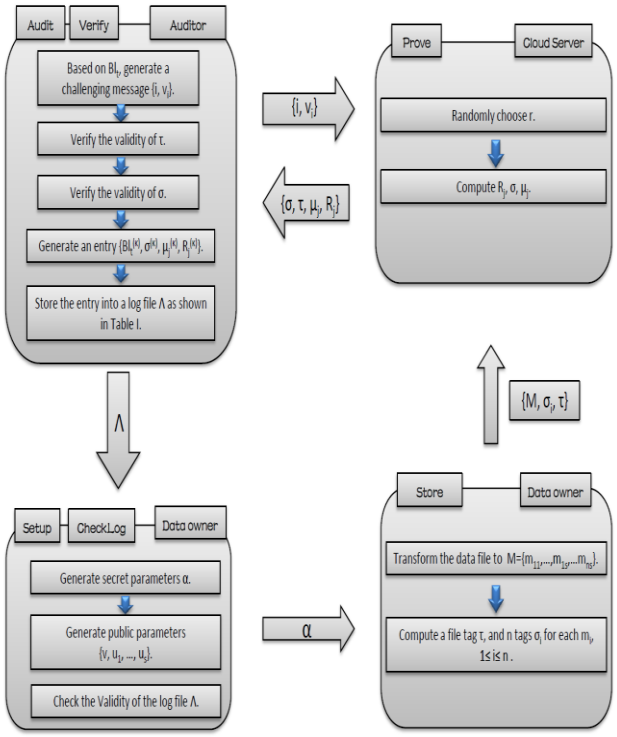

Fig. Execution steps of our scheme

As shown in Fig. 2, the enhanced scheme consists of six algorithms: Setup, Store, Audit, Prove, Verify, Check Log. In our enhanced scheme, the first two algorithms are the same as the SWP. In Audit, the auditor first acquires t Bl based on the current time t and initializes the pseudorandom bit generator

CONCLUSION

In this paper, we propose a novel public verification scheme to resist external adversaries and malicious auditors. The proposed scheme does not require the secure channel between cloud servers and auditors. The proposed scheme is secure against an external active and online adversary, who cannot invalidate the auditor's verification. The proposed scheme is also secure against a malicious auditor, who may fabricate a verification result or collude with the cloud servers to invalidate the proposed scheme. In regards to future work, research efforts can be put into the following areas.

Optimization of Verification Overhead: Most of the existing public verification schemes are based on the public-key cryptosystem. Even the auditor is equipped with a powerful device, the time of verification on the auditor side is asecond (hundreds of millisecond) computation. For these schemes, it is impractical to verify the data integrity by the

auditor with a low-power device. How to reduce the computation from the operations in the public-key cryptography to those in the symmetric-key cryptography while keeping the appealing features of public verification is still an open issue, which deserves further investigation.

Multiple-User and Multiple-Server: In practice, an auditor always serves multiple cloud users and multiple cloud servers. If the auditor handle multiple verification tasks from different users and cloud servers one by one, the verification may incur a huge delay and become a bottleneck in applications. Therefore, how a single auditor can simultaneously handle multiple verification tasks from different users and different cloud servers with high efficiency is worth to be further studied.

New Framework from New Cryptographic Primitives:

With the development of cryptography, lots of new cryptographic primitives are proposed, such as program obfuscation and structure-preserving cryptography. These new cryptographic primitives bring new appealing features and powerful functionalities that cannot be achieved in the past. We believe that combining current public verification technique with these new and powerful cryptographic primitives is promising, which may provide users more feature-rich cloud storage services than ever. This remains an open research issue.

REFERENCES

[1] Shui Yu, YonghongTian, Song Guo, and Dapeng Oliver Wu. "Can We Beat DDos Attacks in Clouds?" IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2245-2254, September, 2014.

[2] Shui Yu, Guojun Wang, and Wanlei Zhou."Modeling Malicious Activites in Cyber Space." IEEE Network, vol. 29, no. 6, pp. 83-87, November/December, 2015.

[3] Shui Yu, Song Guo, and Ivan Stojmenovic. "Fool Me If You Can: Mimicking Attacks and Anti-Attacks in Cyberspace." IEEE Transactions on Computers, vol.64, no. 1, pp. 139-151, January, 2015.

[4] Cong Wang, Sherman S.M. Chow, Qian Wang, KuiRen, and Wenjing Lou."Privacy-Preserving Public Auditing for Secure Cloud Storage." IEEE Transactions on Computers, vol. 62, no. 2, pp. 362-375, February, 2013.

[5] Hongwei Li, Yi Yang, Tom H. Luan, Xiaohui Liang, Liang Zhou, and XuemingShen. "Enabling Fine-grained Multi-keyword Search Supporting Classified Sub-dictionaries over Encrypted Cloud Data." IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 3, pp. 312-325, May/June, 2016.

[6] Hongwei Li, Dongxiao Liu, Yuanshun Dai, and Tom H. Luan. "Engineering Searchable Encryption of Mobile Cloud Networks: When QoE Meets QoP." IEEE Wireless Communications, vol. 22, no. 4, pp. 74-80, August, 2015.

[7] Cong Wang, Qian Wang, KuiRen, and Wenjing Lou. "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing." Proceedings of the 2010 IEEE International Conference on Computer Communications (INFOCOM 2010), IEEE, 2010, pp. 1-9.

[8] ChunxiangXu, Yuan Zhang, Yong Yu, Xiaojun Zhang, and Junwei Wen."An Efficient Provable Secure Public Auditing Scheme for Cloud Storage." KSII Transactions on Internet and Information Systems, vol. 8, no. 11, November, 2014.

[9] Yuan Zhang, ChunxiangXu, Jining Zhao, Xiaojun Zhang, and Junwei Wen. "Cryptanalysis of an Integrity Checking Scheme for Cloud Data Sharing." Journal of Information Security and Applications, vol. 23, pp. 68-73, August, 2015.

[10] Yuan Zhang, ChunxiangXu, Shui Yu, Hongwei Li, and Xiaojun Zhang. "SCLPV: Secure Certificateless Public Verification for Cloud-Based Cyber-Physical-Social Systems Against Malicious Auditors." IEEE Transactions on Computational Social Systems, vol. 2, no. 4, pp. 159-170, December, 2015.

[11] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song."Provable Data Possession at Untrusted Stores." Proceedings of the 2007 ACM SIGSAC Conference on Computer and Communications Security (CCS 2007), ACM, 2007, pp. 598-609.

[12] Solomon GuadieWorku, ChunxiangXu, Jining Zhao, and Xiaohu He."Secure and Efficient Privacy-Preserving Public Auditing Scheme for Cloud Storage."Computers and Electrical Engineering, vol. 40, no. 5, pp.1703-1713, July, 2014.

[13] HovavShacham and Brent Waters."Compact Proofs of Retrievability." Journal of Cryptology, 26.3 (2013): 442-483.

[14] FrederikArmknecht, Jens-Matthias Bohli, Ghassan O. Karame, Zongren Liu, and Christian A. Reuter. "Outsourced Proofs of Retrievability."Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS 2014). ACM, 2014, pp. 831-843.

[15] Dan Boneh, Ben Lynn, and HovavShacham."Short Signatures from the Weil Pairing."Advances in Cryptology -- ASIACRYPT 2001. Springer Berlin Heidelberg, 2001, pp. 514-532.

Authors Profile

Bukya Ramachandra Naickreceived Bachelor of Computer Science degree fromsriVenkateswara University, Tirupati in the Year of 2012-2015. Currently pursuing Master of Computer Applications from Sri Venkateswara University, Tirupati in the Year of 2016-2019. Research interest in the field of Computer Science in the area of Cryptography-Network Security, Cloud Computing and Software Engineering.

Prof Dr S. Ramakrishna, working as a Professor in Dept of Computer Science, Sri Venkateswara University College of Commerce Management and Computer Science, Tirupati, (AP)-India. Received M.Sc, M.Phil, M.Tech (IT) and Doctorate in Computer Science from S.V University, Tirupati, having 27 years experience in teaching field. Additional Assignments Working as Dean of Examinations for S.V University, Worked as Additional Convener for S.V University RESET Examinations, Worked as Coordinator for M.Sc Computer Science, Worked as BoS Chairman in Computer Science. Research Papers Published in National & International Journals :99, Total Number of Conferences participated :33, Total number of Books Published:7, Total number of Training Programs Attended : 3, Total number of Orientation & Refresher Courses Attended : 4.Number of research degrees awarded under my guidance :- M.Phil: 20,Ph.D:20.