ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 17, March 2016

PUBLIC INTEGRITY VERIFICATION SCHEME AND USER LEVEL INTEGRITY IN DYNAMIC CLOUD ENVIRONMENT

LAKSHMIPRIYA.T [1], MATHESWARAN.V [2]

1. P.G. Student, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India
2. Asst.Professor, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India

Abstract : Cloud computing moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. In this work, we study the problem of ensuring the integrity of data storage in cloud computing. To reduce the computational cost at user side during the integrity verification of their data, the notion of public verifiability has been proposed. However, the challenge is that the computational burden is too huge for the users with resource-constrained devices to compute the public authentication tags of file blocks. To tackle the challenge, we propose OPoR, a new cloud storage scheme involving a cloud storage server and a cloud audit server, where the latter is assumed to be semi-honest. In particular, we consider the task of allowing the cloud audit server, on behalf of the cloud users, to pre-process the data before uploading to the cloud storage server and later verifying the data integrity. OPoR outsources and offloads the heavy computation of the tag generation to the cloud audit server and eliminates the involvement of user in the auditing and in the pre-processing phases. Furthermore, we strengthen the proof of retrievability (PoR) model to support dynamic data operations, as well as ensure security against reset attacks launched by the cloud storage server in the upload phase.

Keywords: - cloud storage, proof-ofretrievability, cloud audit server, cloud storage server.

I. INTRODUCTION

Cloud computing is a computing technology which provides different types of services through Internet. It shares the software and hardware resources, and provide resources to user's computer or mobile device. Cloud service provider's offers three important services. They are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Cloud computing has four models. They are ,

Public cloud: Public clouds are prepared to be available to the general public. Generally , public cloud providers like Microsoft and offer is access over the internet. In this model, customers doesn't have any visibility or control over where the infrastructure is.

Private cloud: It is a cloud infrastructure which is designed to a particular organization. It doesn't share with other organizations, whether managed internally or externally by third-party, and it can be hosted externally or internally.

Community Cloud: here the cloud is a multi-tenant cloud service model which is shared among several organization. It is controlled and managed commonly by all the participating organizations.

Hybrid cloud: It is a composition of two or more clouds (private, public or community cloud).

Cloud storage service is the most common and popular service among many cloud services (e.g. Google Drive, Dropbox, etc...) for general users. They have a bottleneck in local storage space because there are more and more users save their data in cloud storage, so cloud storage service has high capacity which solves user's difficulty in storing and retrieving data. Besides, cloud storage service provides high capacity space in order to achieve ubiquitous service, it also provides access to cloud services from web services or applications which utilize the application programming interface by mobile devices (e.g. laptop, table computer and smart phones). Even though the cloud storage service has many advantages, it brings a lotof challenging issues which include efficiency and security. Christo Ananth et al. [3] discussed about a method, End-to-end inference to diagnose and repair the data-forwarding failures, our optimization goal to minimize the faults at minimum expected cost of correcting all faulty nodes that cannot properly deliver data. First checking the nodes that has the least checking cost does not minimize the expected costin fault localization. We construct a potential function for identifying the candidate nodes, one of which should be first checked by an optimal strategy. We proposes efficient inferring approach to the node to be checked in large-scale networks**.**

Integrity, is a one which is used to ensure that the data stored is as it is, there is no modification have been done in it. The data can modified and accessed only by the authorized users. Data Integrity is a main challenging issue in cloud computing. In cloud computing users store their data. Then the users relies on cloud servers for storage and maintenance. But such hope may also fail sometimes. Because the servers may also misuse or delete the rarely accessed data of user's in order to maintain their reputation. So integrity verification is main concern in cloud computing. For this purpose we have two integrity verification methods. They are private auditing and public auditing. In private auditing the authorized one only can audit the users data whereas in public auditing anyone can perform the auditing task. In my paper I am going to introduce public auditing with privacy preservation. So anyone can audit data correctness of my data without accessing the actual file content.

The Proof of Retrievability (PoR) is an archive that provides a concise proof that the user can retrieve the target file. PoR is an important tool for semi-trusted online archives. The users can view their file in the archive but they cannot modify the data in the file. The goal of a PoR is to accomplish these checks without users having to download the files themselves. Also in a PoR the cloud storage must prove to a verifier that is the client that it is storing all the clients' data. Although PoR provides many advantages some disadvantages are also found wit PoR. The users or the clients cannot modify their data in the file. Some security problems are also found and also the computational cost is found to be high with PoR. Also some integrity problems are found. To tackle all the challenges faced by PoR a new scheme OPoR (Outsourced Proof of Retrievability) is used. It includes two independent servers the cloud storage server and the cloud audit server. The cloud audit server has some additional capabilities that the clients do not have and this is also responsible for preprocessing the data instead of the clients. By using OpoR dynamic data operations can be performed. And all the security concerns are avoided.

II. RELATED WORKS

There are many approaches has been proposed for data integrity verification in cloud computing . In 2007, the provable data possession (PDP) model is proposed by Ateniese et al. [1]. It uses public auditability and ensures the possession of user's data on untrusted storage. Here they were used RSA based homomorphic verifiable tags in order to audit the outsourced data. Their scheme provides both block less verification and public verifiability at the same time. Even though, Ateniese et al.'s scheme cannot support verification of dynamic data because their scheme only considers the static data situation where the client stores outsourced data and will not modify it. In order to overcome this, Ateniese et al. [2] proposed a scalable PDP scheme in 2008 to improve dynamic data verification. However, their scheme cannot support fully dynamic data. Because their scheme cannot support block insertions and only allows simple block operation which performs partially dynamic data like block level modification and block level deletion. A challenge-response protocol is proposed by Wang et al. [9] which is used to determine data correctness and locate possible errors. But, their scheme supports only partially dynamic data operation. Christo Ananth et al. [6] discussed about a method, Wireless sensor networks utilize large numbers of wireless sensor nodes to collect information from their sensing terrain. Wireless sensor nodes are battery-powered devices. Energy saving is always crucial to the lifetime of a wireless sensor network. Recently, many algorithms are proposed to tackle the energy saving problem in wireless sensor networks. There are strong needs to develop wireless sensor networks algorithms with optimization priorities biased to aspects besides energy saving. In this project, a delay-aware data collection network structure for wireless sensor networks is proposed based on Multi hop Cluster Network. The objective of the proposed network structure is to determine delays in the data collection processes. The path with minimized delay through which the data can be transmitted from source to destination is also determined. AODV protocol is used to route the data packets from the source to destination.

Juels and Kaliski [4] introduced the proof of retrievability (POR) model, in which the spot-checking and error-correcting codes can make sure the possession of data files and retrievability of it on remote archive service systems. However, their scheme only suitable for static data storage because the number of queries can performed by a client is fixed a priori and embedding special blocks which prevent the development of dynamic data updates. Shacham and Waters [7] proposed an improved POR scheme. It uses BLS signature in order to replace RSA-based signature and also to reduce the proof size. They were used secure random oracle model with public verifiable homomorphic linear authenticators which are built from BLS signature. They proved that it is secure in a polynomial extraction algorithm to reveal messages. However, they were only considered static data operation. In order to satisfy public verification and dynamic data Wang et al. [10] proposed a new scheme.

In this scheme they improves an index of data block which can support fully dynamic data. They also extended their scheme to support batch auditing which is used to improve efficiency. Wang et al. [8] pointed out that Wang et al.'s scheme has data privacy issues because the TPA can get the client's data information. So, they use a random mask technology in order to avoid TPA learning knowledge on every verification process. Li et al. [5] considered that the client's resource-constrained device which is simple and lightweight. Therefore, they propose a new scheme which delegate TPA to execute high computing process and solve the client's overhead

2.1 EXISTING SYSTEM

Although having appealing advantages as a promising service platform for the Internet, this new data storage paradigm in "Cloud" brings many challenging issues which have profound influence on the usability, reliability, scalability, security, and performance of the overall system. One of the biggest concerns with remote data storage is that of data integrity verification at un trusted servers. For instance, the storage service provider may decide to hide such data loss incidents as the Byzantine failure from the clients to maintain a reputation. What is more serious is that for saving money and storage space the service provider might deliberately discard rarely accessed data files which belong to an ordinary client. Considering the large size of the outsourced electronic data and the client's constrained resource capability, the core of the problem can be generalized as how can the client find an efficient way to perform periodical integrity verification without the local copy of data files.

2.1.1 Disadvantages of Existing System

1. Less security

2. Less performance

2.2 PROPOSED SYSTEM

We propose OPoR, a new PoR scheme with two independent cloud servers. Particularly, one server is for auditing and the other for storage of data. The cloud audit server is not required to have high storage capacity. Different from the previous work with auditing server and storage server, the user is relieved from the computation of the tags for files, which is moved and outsourced to the cloud audit server. Furthermore, the cloud audit server also plays the role of auditing for the files remotely stored in the cloud storage server. We develop a strengthened security model by considering the reset attack against the storage server in the upload phase of an integrity verification scheme. It is the first PoR model that takes reset attack into account for cloud storage system. We present an efficient verification scheme for ensuring remote data integrity in cloud storage. The proposed scheme is proved secure against reset attacks in the strengthened security model while supporting efficient public verifiability and dynamic data operations simultaneously.

2.2.1 Advantages of Proposed System

1. More security

2. More performance

III. SYSTEM ARCHITECTURE

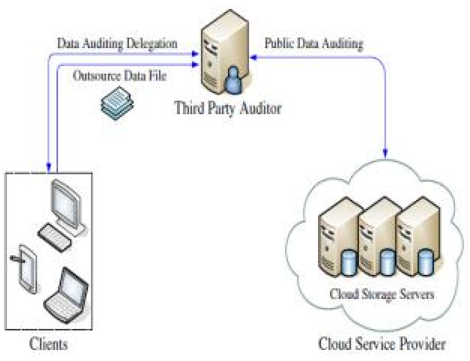

Fig. 1: Cloud data storage architecture

We propose OPoR, a new PoR scheme with two independent cloud servers. Particularly, one server is for auditing and the other for storage of data. The cloud audit server is not required to have high storage capacity. Different from the previous work with auditing server and storage server, the user is relieved from the computation of the tags for files, which is moved and outsourced to the cloud audit server. Furthermore, the cloud audit server also plays the role of auditing for the files remotely stored in the cloud storage server. We develop a strengthened security model by considering the reset attack against the storage server in the upload phase of an integrity verification scheme. It is the first PoR model that takes reset attack into account for cloud storage system. We present an efficient verification scheme for ensuring remote data integrity in cloud storage. The proposed scheme is proved secure against reset attacks in the strengthened security model while supporting efficient public verifiability and dynamic data operations simultaneously. Representative network architecture for cloud data storage is illustrated in Figure 1. Three different network entities can be identified as follows:

Client: an entity that has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations.

Cloud Storage Server (CSS): an entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain client's data. The CSS is required to provide integrity proof to the clients or cloud audit server during the integrity checking phase.

Cloud Audit Server (CAS): a TPA, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request. In this system, the cloud audit server also generates all the tags of the files for the users before uploading to the cloud storage server.

In the cloud paradigm, by putting the large data files on the remote servers, the clients can be relieved of the burden of storage and computation. As clients no longer possess their data locally, it is of critical importance for the clients to ensure that their data are being correctly stored and maintained. That is, clients should be equipped with certain security means so that they can periodically verify the correctness of the remote data even without the existence of local copies. In case that clients do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the monitoring task to a trusted cloud audit server of their respective choices.

A. Design Goals

Our design goals can be summarized as the following: (1) Public verifiability: to allow anyone, not just the clients originally stored the file, to have the capability to verify correctness of the remotely stored data; (2) Low computation overhead at the client side: to upload data to the cloud server while supporting verifiability, the data owner does not have heavy additional computation; (3) Dynamic data operation support: to allow the clients to perform block-level operations on the data files while maintaining the same level of data correctness assurance; (4) Stateless verification: to eliminate the need for state information maintenance at the verifier side between audits and throughout the long term of data storage. This is also the basic requirement for achieving public verifiability. In particular, we aim to achieve enhanced security against reset attacks in our construction.

B.Modules Description

- **Client module:** an entity that has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations.
- **Cloud Storage Server (CSS) module:** an entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain client's data. The CSS is required to provide integrity proof to the clients or cloud audit server during the integrity checking phase.
- **Cloud Audit Server (CAS) module:** a TPA, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request. In this system, the cloud audit server also generates all the tags of the files for the users before uploading to the cloud storage server. The basic goal of PoR model is to achieve proof of retrievability. Informally, this property ensures that if an adversary can generate valid integrity proofs of any file F for a non-negligible fraction of challenges, we can construct a PPT machine to extract F with overwhelming probability. It is formally defined by the following game between a challenger C and an adversary A, where C plays the role of the audit server (the client) and A plays the role of the storage server:
- **Setup Phase**: The challenger C runs the Setup algorithm to generate its key pair (pk, sk), and forwards pk to the adversary A.
- **Upload Phase:** C initiates an empty table called Rlist. A can adaptively query an upload oracle with reset capability as follows: – Upload: When a query on a file F and a state index i comes, C checks if there is an entry (i, ri) in the R-list. If the answer is yes, C overwrites ri onto its random tape; otherwise, C inserts (i, ri) into R-list where ri is the content on its random tape. Then C runs (F∗, t) ← Upload(sk, F; ri), and returns the stored file F∗ and the file tag t. Here Upload(• ; ri) denotes an execution of the upload algorithm using randomness ri.
- **Challenge Phase:** A can adaptively make the following two kinds of oracle queries: IntegrityVerify: When a query on a file tag t comes, C runs the integrity verification protocol Integrity Verify {A C(pk, t)} with A.

  Update: When a query on a file tag ˆt and a data operation request "update" comes, C runs the update protocol Update{A C(sk, ˆt, update)} with A.

IV.CONCLUSION AND FUTURE WORK

This paper proposes OPoR, a new proof of retrievability for cloud storage, in which a trustworthy audit server is introduced to preprocess and upload the data on behalf of the clients. In OPoR, the computation overhead for tag generation on the client side is reduced significantly. The cloud audit server also performs the data integrity verification or updating the outsourced data upon the clients' request. Besides, we construct another new

PoR scheme proven secure under a PoR model with enhanced security against reset attack in the upload phase. The scheme also supports public verifiability and dynamic data operation simultaneously. There are several interesting topics to do along this research line. For instance, we can (1) reduce the trust on the cloud audit server for more generic applications, (2) strengthen the security model against reset attacks in the data integrity verification protocol, and (3) find more efficient constructions requiring for less storage and communication cost. We leave the study of these problems as our future work.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in *CCS ’07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2007, pp. 598–609.

[2] A. Juels and B. S. K. Jr., “Pors: proofs of retrievability for large files,” in *CCS ’07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA:ACM, 2007, pp. 584–597.

[3] Christo Ananth, Mary Varsha Peter, Priya.M., Rajalakshmi.R., Muthu Bharathi.R., Pramila.E., “Network Fault Correction in Overlay Network through Optimality”, International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Volume 2, Issue 8, August 2015, pp: 19-22

[4] K. D. Bowers, A. Juels, and A. Oprea, “Proofs of retrievability: theory and implementation,” in *Proceedings of CCSW 2009*. ACM, 2009, pp. 43–54.

[5] M. Naor and G. N. Rothblum, “The complexity of online memory checking,” *J. ACM*, vol. 56, no. 1, pp. 2:1–2:46, Feb. 2009. [Online]. Available: http://doi.acm.org/10.1145/1462153. 1462155

[6] Christo Ananth, T.Rashmi Anns, R.K.Shunmuga Priya, K.Mala, “Delay-Aware Data Collection Network Structure For WSN”, International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Special Issue 2 - November 2015, pp.17-21

[7] M. A. Shah, R. Swaminathan, and M. Baker, “Privacy-preserving audit and extraction of digital contents,” Cryptology ePrint Archive, Report 2008/186, 2008, http://eprint.iacr.org/.

[8] A. Oprea, M. K. Reiter, and K. Yang, “Space-efficient block storage integrity,” in *In Proc. of NDSS 2005*, 2005.

[9] T. S. J. Schwarz and E. L. Miller, “Store, forget, and check: Using algebraic signatures to check remotely administered storage,” in *ICDCS ’06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2006.

[10] Q. Wang, K. Ren, S. Yu, and W. Lou, “Dependable and secure sensor data storage with dynamic integrity
assurance,” *ACM Transactions on Sensor Networks*, vol. 8,no. 1, pp. 9:1–9:24, Aug. 2011. [Online]. Available: http://doi.acm.org/10.1145/1993042.1993051

[11] L. V. M. Giuseppe Ateniese, Roberto Di Pietro and G. Tsudik, “Scalable and efficient provable data possession,” in *International Conference on Security and Privacy in Communication Networks(SecureComm 2008)*, 2008.

[12] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for data storage security in cloud computing,” in *INFOCOM*, 2010, pp. 525–533.

[13] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, “Dynamic audit services for integrity verification of outsourced storages in clouds,” in *SAC*, 2011, pp. 1550–1557.

[14] Q. Zheng and S. Xu, “Fair and dynamic proofs of retrievability,” in *CODASPY*, 2011, pp. 237–248.

[15] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, “Fine-grained access control system based on attribute-based encryption,” *ESORICS*, 2013.

[16] J. Li and K. Kim, “Hidden attribute-based signatures without anonymity revocation,” *Information Sciences*, vol. 180, no. 9, pp. 1681–1689, 2010.

[17] J. Li, C. Jia, J. Li, and X. Chen, “Outsourcing encryption of attribute-based encryption with mapreduce,” *ICICS*, 2012.

[18] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, “New algorithms of outsourcing modular exponentiations,” *ESORICS*, pp. 541–556, 2012.

[19] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, “Cooperative provable data possession for integrity verification in multicloud storage,” *IEEETrans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2231–2244, 2012.

[20] H. Xiong, X. Zhang, D. Yao, X. Wu, and Y. Wen, “Towards endto-end secure content storage and delivery with public cloud,” in *CODASPY*, 2012, pp. 257–266.

[21] Q. Zheng and S. Xu, “Secure and efficient proof of storage with deduplication,” in *CODASPY*, 2012, pp. 1–12.

[22] C. Wang, Q. Wang, and K. Ren, “Ensuring data storage security in cloud computing,” in *Proceedings of IWQoS 2009*, Charleston, South Carolina, USA, 2009.

[23] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, “Dynamic provable data possession,” Cryptology ePrint Archive, Report 2008/432, 2008, http://eprint.iacr.org/.

[24] X. Lei, X. Liao, T. Huang, H. Li, and C. Hu, “Outsourcinglarge matrix inversion computation to a public cloud,,” in *IEEETransactions on Cloud Computing*, 2013, pp. vol. 1, no. 1.