ISSN: 0975-766X
CODEN: IJPTFI
Research Article

Available Online through
www.ijptonline.com

AN REVIEW AND DESIGN OF PRIVILEGED DATA SHARING IN BATCH AUDIT CLOUD

Senthilkumar.R[1], Dr.Geetha B.G.[2]
[1]Research Scholar/Anna university Chennai and Assistant Professor/ CSE,SVHEC,Gobi
[2]Professor and Head/ K.S.Rangasamy College of technology (Autonomous), Tiruchengode.
Email: yoursrsk@gmail.com

Received on: 10-03-2017 *Accepted on: 02-04-2017*

Abstract

Cloud storage system is consisting of storage servers with long-term storage services over the World on the internet. Cloud storage has the possible of providing purely distributed storage services. Since cloud can integrate servers and clusters that are disseminated all over the world and offered by different service providers into one virtualized atmosphere. Cloud data storage belongs to IaaS. This allows the Clients to move their data from local computing systems to the remote. A general sharing scheme protects data privacy but also restricts the functionality of the cloud data storage system because at some operations are supported to over encrypted data. Cloud computing provides a reasonable and proficient solution for data sharing group resource among cloud users. In cloud data storage system, the Clients store their data in the cloud and no longer possess the data locally. After data goes into the cloud, the Client loses control over it. If such data storage is vulnerable to attacks, in which the adversary can modify or delete the data or inject polluted data into the servers or may be access the data. This Paper first finds the difficulties and probable security problems of direct extensions with fully lively data updates from prior works and then shows how to build an elegant verification scheme for a seamless combination of these two outstanding features in this procedure design.

Keywords - Cloud Computing, virtualization, Data Sharing, TPA.

1. Introduction

Checking the authenticity of data has emerged as a critical issue in storing data on untreated cloud servers. It arises in peer-to-peer storage systems, network file systems long term archives, web-service object stores, and database systems. Such systems block storage servers from misbehaving or editing data by providing authenticity checks when accessing data. Archival storage requires guarantees about the authenticity of data on storage, namely that storage servers possess data. It is insufficient to find that data have been changed or removed when accessing the data because it may be too late to recover lost or damaged data. Archival storage servers retain tremendous amounts of data. They also hold data for a long time during which there may be exposure to data loss from administration errors as the physical implementation of storage, evolves, data migration to new systems, and changing memberships in peer-to-peer systems.

Although envisioned as a promising service platform for the Internet, this new data storage paradigm in “Cloud” brings about many challenging design issues which have a profound influence on the security and performance of the overall system. One of the biggest concerns with cloud data storage is that of data integrity verification on untrusted servers. The cloud, which meets Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. What is more problematic is that for saving money and storage space the service provider might neglect to keep or deliberately remove rarely accessed data files which belong to an ordinary user. Consider a large amount 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 checks without the local copy of data files.

To solve this problem, many methods are proposed under different systems and security models. A great effort is made to define solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data. Considering the role of the verifier in the model, all the methods proposed before falling into two categories: private verifiability and public verifiability. Although schemes with private verifiability can get higher method efficiency, public verifiability allows anyone, not just the data owner, to challenge the cloud server for the correctness of data storage while keeping no private information. Clients are able to delegate the evaluation of the service performance to an independent third party auditor (TPA), without the devotion of their computation resources.

The users themselves are unreliable or cannot afford the overhead of performing frequent integrity checks. It seems more rational to equip the verification protocol with public verifiability, which is expected to play a more important role in achieving economies of scale for Cloud Computing. For efficiency consideration, the outsourced data themselves should not be needed by the verifier for the verification purpose.

Another major concern among existing designs is that of supporting dynamic data operations for cloud data storage applications. In Cloud, the remotely stored electronic data might not only be accessed but also updated by the users. In the context of remote data storage mainly concentrates on static data files and the importance of this dynamic data updates has received limited attention in the data possession applications so far. The direct extension of the currently provable data possession (PDP) or proof of retrievability (PoR) method to support data dynamics may lead to security loopholes. Although there are many difficulties faced by scholars', it is well believed that supporting dynamic data operation can be of vital importance to the practical application of storage outsourcing services.

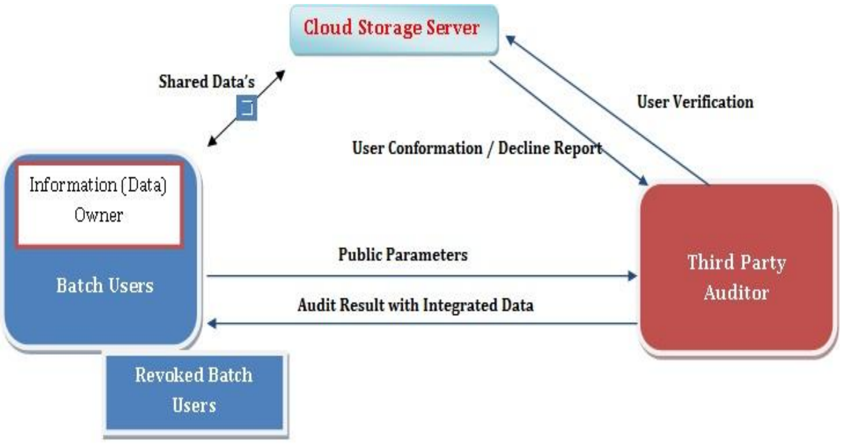

Fig. 1: Cloud data storage architecture.

This Paper involvement is summarized as follows:

(1) Propose a general formal PoR model with public verifiability for cloud data storage, in which block less verification is achieved;

(2) Design the proposed PoR construction with the function of supporting for fully dynamic data operations, especially to support block addition, which is missing in most existing schemes;

(3) To prove the security of our proposed work and justify the performance of our method through concrete implementation and comparisons.

2. Overview of Proposed Methodology

This paper introduces audit system architecture for outsourced data in clouds as shown in Figure. In this architecture consider a data storage service involving four entities: data owner (DO), who has a large amount of data to be stored in the cloud; cloud service provider (CSP), who provides data storage service and has enough storage space and computation resources; third party auditor (TPA), who has capabilities to manage or monitor the outsourced data under the delegation of data owner; and authorized applications (AA), who have the right to access and manipulate stored data. Finally, application users can enjoy various cloud application services via these authorized applications. A user, assume that TPA is reliable and independent through the following audit functions: TPA should able to make regular checks on integrity and availability of delegated data at appropriate intervals; TPA should able to organize, manage, and maintain the outsourced data instead of a data owners, and support the dynamic data operations for authorized applications; and TPA should take the evidences for disputes about the inconsistency of data in terms of authentic records for all data operations to realize the functions, our audit service is comprised of three processes: Tag Generation: the client (data owner) uses the secret key to the pre-process file, which consists a collection of n blocks, generates the set of public verification parameters (PVP) and index-hash table (IHT) that are stored in TPA, transmits the file and some verification tags in CSP, and may delete its local copy; Periodic Sampling Audit: by using an interactive proof protocol of retrievability, TPA (or other applications) issues a "Random Sampling" challenge to audit the integrity and availability of the outsourced data in terms of the verification information stored in TPA.

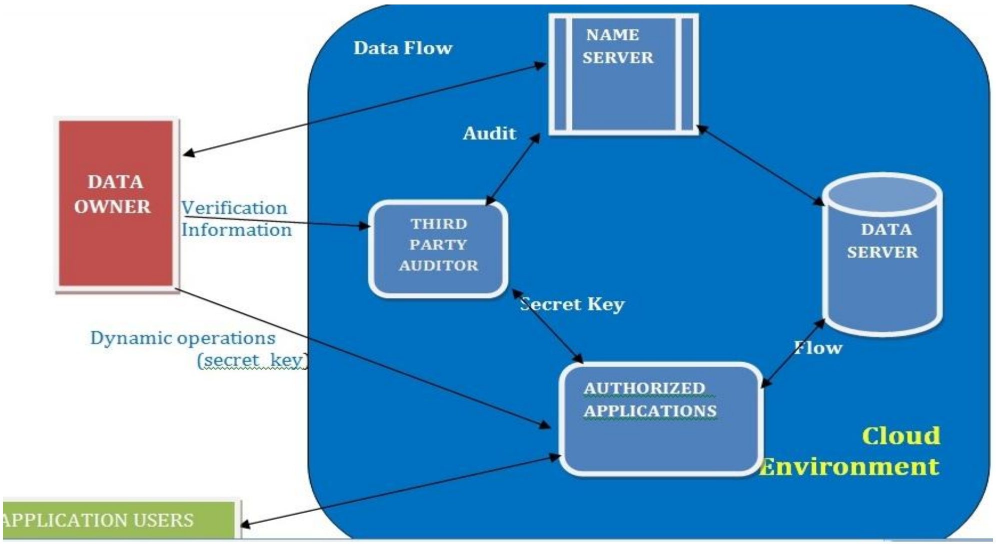

Fig. 2: The audit system architecture.

Audit of Dynamic Operations: An authorized application holds the data owner's secret key SK, can manipulate an outsourced data and update associated index hash table (IHT) stored in TPA. The privacy of SK and the checking algorithm ensure that storage server cannot cheat the authorized applications and forge the valid audit records. In general, the authorized applications should be cloud application services inside clouds for various application purposes, they must be specifically authorized by data owners for manipulating the outset data. Since the acceptable operations require that authorized applications must present authentication information for TPA, any unauthorized modifications of data will be detected in audit processes or verification processes. Based on this kind of strong authorization-verification mechanism, neither assumes that CSP is trusted to guarantee the security of stored data, nor assume that a date owner has the capability to collect the evidence of CSP's faults after errors have been found. The ultimate goal of the audit infrastructure is enhance the credibility of cloud storage services, but not to increase the data owner's burden and overheads. For this purpose, TPA should be constructed in clouds and maintained by a cloud storage provider (CSP). In order to ensure the trust and security, TPA must be secure enough to resist malicious attacks, and it also should be strictly controlled to prevent unauthorized access even for internal members in clouds.

A more practical way is that the TPA in the clouds should be mandated by a trusted third party (TTP). This mechanism not only improves the performance of audit services, but also provides the data owner with a maximum access transparency. This means that data owners are entitled to utilize the audit service without further costs besides storing a secret-key and some secret information. The above processes involve some procedures: KeyGen, TagGen, Update, Delete, and Insert algorithms, as well as an interactive proof protocol of retrievability. In order to improve the security and performance, we make use of following techniques to construct corresponding algorithms and protocols.

Proposed System Contributions

- Propose an efficient public integrity auditing scheme for cloud data sharing that supports multiple writers.
- To novel design on polynomial-based authentication tags, empower the cloud to aggregate authentication tags from multiple writers.
- A constant size of integrity proof information and a constant number of computational operations are needed for the verifier.
- The novel proxy authentication tag update technique, the scheme allows secure delegation of user revocation operations to the cloud.

The proposed scheme allows aggregation of integrity auditing operations for multiple tasks (files) through batch integrity Technique.

3. Conclusion and Prospect Works

Security of Cloud Computing is a part of ongoing research. The proposed scheme is featured by salient properties of public integrity auditing and constant computational cost on the user side. The user achieves this through design on polynomial-based virtualization endorsement tags which allows aggregation of tags of dissimilar data blocks. Cloud user data application has many issues have identified the security and the cloud services are available to achieve security with the many techniques and methods. Based on the user requirements, the user selects one of the security services tools as proposed.

Once the data is classified and tagged, then the level of security associated with this specific tagged data element can be applied.

The level of data security is included confidentiality, encryption, integrity, and storage etc. This paper has used to understand the security issues and given one of the solutions in system architecture. In Feature a secure and trusted solution is the requirement that needs to be focused on the cloud computing infrastructure.

4. References

1. Erica Sousa, Fernando Lins and Eduardo Tavares,Paulo Cunha and Paulo Maciel, “Modeling Approach for Cloud Infrastructure Planning Considering Dependability and Cost Requirements”, IEEE Transactions On Systems, Man, And Cybernetics: Systems, Vol. 45, No. 4, April.
2. Kejiang Ye, Zhaohui Wu, Chen Wang, Bing Bing Zhou, Weisheng Si, Xiaohong Jiang and Albert Y. Zomaya, “Profiling-Based Workload Consolidation and Migration in Virtualized Data Centers”, IEEE Transactions On Parallel And Distributed Systems, Vol. 26, No. 3, March 2015.
3. J. Yuan and S. Yu, “Efficient public integrity checking for cloud data sharing with multi-user modification,” in Proc. 33rd Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM), Toronto, ON, Canada, Apr./May 2014, pp. 2121–2129.
4. Yajuan Tang, Xiapu Luo, Qing Hui and Rocky K. C. Chang, “Modeling the Vulnerability of Feedback-Control Based Internet Services to Low-Rate DoS Attacks”, IEEE Transactions On Information Forensics And Security, Vol. 9, No. 3, March 2014.
5. Kui Xu, Patrick Butler, Sudip Saha and Danfeng (Daphne) Yao, “DNS for Massive-Scale Command and Control”, IEEE Transactions On Dependable And Secure Computing, Vol. 10, No. 3, May June 2013.
6. Aida Ghazizadeh,”Cloud Computing Benefits and Architecture in E-Learning,” IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education, pp.199-201, June 2012.
7. Paul S Wooley, “Identifying Cloud Computing Security Risks”, University of Oregon Libraries, February 2011.
8. Qian Wang, Cong Wang ,” Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing” in March 2011.
9. Masayuki Okuhara et al, “Security Architecture for Cloud Computing”, FUJITSU Sci. Tech. J., Vol. 46, No.4, pp. 397-402 ,October 2010
10. Daniele Catteddu, Giles Hogben, “Cloud Computing Benefits, risks and recommendations for information security”, May 2009.