ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

Privacy Preserving and Efficient Access Control Using Public Verifiable System in Cloud Computing

S.Md.Tasneem Yasmeen[1], P. Vasanthi [2]

M.Tech Student, Dept. of CSE, GATES Engineering College, Affiliated to JNTUA , Andhra Pradesh, India [1]

Assistant Professor, Dept. of CSE, GATES Engineering College, Affiliated to JNTUA, Andhra Pradesh, India[2]

ABSTRACT: Cloud data storage service involving three different entities the cloud user(CU),the cloud server (CS), the third party verifier (TPV), who has experience and capabilities that cloud users do not have and is trusty to assess the cloud storage service reliable on behalf of the user upon request. Cloud users may resort to TPV for ensuring the storage integrity of their outsourced data, the existence of a semi-trusted CS as does. Namely, in most of your time it behaves properly and doesn't deviate from the prescribed protocol execution. However, for their advantages the CS may neglect to stay or deliberately delete rarely accessed data files that belong to standard cloud users. Moreover, the CS could plan to hide the data corruptions caused by server hacks or failures to keep up reputation. We have a tendency to assume the TPV, who is within the business of verifying, is reliable and independent, and so has no incentive to interact with either the CS or the users during the verifying process. However, it harms the user if the TPV may learn the outsourced data.

Presenting our public verifying scheme which provides a whole outsourcing solution of data – not only the data itself, however additionally its integrity checking schemes within the context of remote data integrity and adapt the framework for our privacy-preserving public verifying system. Privacy-preserving public verifying system for data storage security in Cloud Computing. we tend to utilize the homomorphic linear appraiser and random masking to ensure that the TPV wouldn't learn any knowledge concerning the data content hold on the cloud server throughout the efficient verifying method.

KEYWORDS: Integrity auditing, public verifiability, dynamic update, arbitration, fairness

I.INTRODUCTION

Cloud computing is the one of the important concept of Information Technology. The cloud computing provides lot of benefits to IT as they are pay per use, reduce the infrastructure technology, stream line process, improve accessibility and improve flexibility. The main aspect of cloud computing is the data has centralized or outsourced to the cloud. From the user's point of view including both individuals and IT environment, they can store the remote data on cloud that has the advantage like without overhead of storage maintenance, global data access with not dependent of geographical locations. The cloud computing provide these benefits. In this cloud storage is consists of the cloud service provider(CSP) as like as cloud server or server for providing services for the client when they require and also separate the administrative entities from the outsource data. The outsource data is actually user's ultimate control over their data. The cloud storing data integrity is having risk.

First of all, earlier auditing schemes usually require the CSP to generate a deterministic proof by accessing the whole data file to perform integrity check, e.g., schemes use the entire file to perform modular exponentiations. Such plain solutions incur expensive computation overhead at the server side, hence they lack efficiency and practicality when dealing with large-size data. Represented by the "sampling" method in "Proofs of Retrievability" (PoR) model and "Provable Data Possession" (PDP) model, later schemes tend to provide a probabilistic proof by accessing part of the file, which obviously enhances the auditing efficiency over earlier schemes.

DOI: 10.15680/IJIRCCE.2017. 0510047

Secondly, some auditing schemes provide private verifiability that require only the data owner who has the private key to perform the auditing task, which may potentially overburden the owner due to its limited computation capability. Ateniese el al. were the first to propose to enable public verifiability in auditing schemes. In contrast, public auditing schemes allow anyone who has the public key to perform the auditing, which makes it possible for the auditing task to be delegated to an external third party auditor (TPA). A TPA can perform the integrity check on behalf of the data owner and honestly report the auditing result to him.

Thirdly, PDP and PoR intend to audit static data that are seldom updated, so these schemes do not provide data dynamics support. But from a general perspective, data update is a very common requirement for cloud applications. If auditing schemes could only deal with static data, their practicability and scalability will be limited. On the other hand, direct extensions of these static data oriented schemes to support dynamic update may cause other security threats.

.

II. LITERATURE SURVEY

1. Y. Deswarte, J.-J. Quisquater, and A. Sa¨ıdane,

This paper analyzes the problem of checking the integrity of files stored on remote servers. Since servers are prone to successful attacks by malicious hackers, the result of simple integrity checks run on the servers cannot be trusted. Conversely, downloading the files from the server to the verifying host is impractical. Two solutions are proposed, based on challenge-response protocols.

2. D. L. Gazzoni Filho and P. S. L. M. Barreto

We observe that a certain RSA-based secure hash function is homomorphic. We describe a protocol based on this hash function which prevents ‘cheating’ in a data transfer transaction, while placing little burden on the trusted third party that oversees the protocol. We also describe a cryptographic protocol based on similar principles, through which a prover can demonstrate possession of an arbitrary set of data known to the verifier. The verifier isn’t required to have this data at hand during the protocol execution, but rather only a small hash of it. The protocol is also provably as secure as integer factoring.

3. A. Juels and B. S. Kaliski Jr

In this paper, we define and explore proofs of retrievability (PORs). A POR scheme enables an archive or back-up service (prover) to produce a concise proof that a user (verifier) can retrieve a target file F, that is, that the archive retains and reliably transmits file data sufficient for the user to recover F in its entirety.

A POR may be viewed as a kind of cryptographic proof of knowledge (POK), but one specially designed to handle a large file (or bit string) F. We explore POR protocols here in which the communication costs, number of memory accesses for the prover, and storage requirements of the user (verifier) are small parameters essentially independent of the length of F. In addition to proposing new, practical POR constructions, we explore implementation considerations and optimizations that bear on previously explored, related schemes.

In a POR, unlike a POK, neither the prover nor the verifier need actually have knowledge of F. PORs give rise to a new and unusual security definition whose formulation is another contribution of our work.

We view PORs as an important tool for semi-trusted online archives. Existing cryptographic techniques help users ensure the privacy and integrity of files they retrieve. It is also natural, however, for users to want to verify that archives do not delete or modify files prior to retrieval. The goal of a POR is to accomplish these checks without users having to download the files themselves. A POR can also provide quality-of-service guarantees, i.e., show that a file is retrievable within a certain time bound

III. EXISTING SYSTEM

First of all, earlier auditing schemes usually require the CSP to generate a deterministic proof by accessing the whole data file to perform integrity check.

Secondly, some auditing schemes provide private verifiability that require only the data owner who has the private key to perform the auditing task, which may potentially overburden the owner due to its limited computation capability.

Thirdly, PDP and PoR intend to audit static data that are seldom updated, so these schemes do not provide data dynamics support. But from a general perspective, data update is a very common requirement for cloud applications.

IV.PROPOSED WORK

This problem is addressed by differentiating between tag index (used for tag computation) and block index (indicate block position), and rely an index switcher to keep a mapping between them. Upon each update operation, we allocate a new tag index for the operating block and update the mapping between tag indices and block indices. Such a layer of indirection between block indices and tag indices enforces block authentication and avoids tag re-computation of blocks after the operation position simultaneously. As a result, the efficiency of handling data dynamics is greatly enhanced.

Furthermore and important, in a public auditing scenario, a data owner always delegates his auditing tasks to a TPA who is trusted by the owner but not necessarily by the cloud.

To address the fairness problem in auditing, we introduce a third-party arbitrator(TPAR) into our threat model, which is a professional institute for conflicts arbitration and is trusted and payed by both data owners and the CSP. Since a TPA can be viewed as a delegator of the data owner and is not necessarily trusted by the CSP, we differentiate between the roles of auditor and arbitrator. Moreover, we adopt the idea of signature exchange to ensure metadata correctness and provide dispute arbitration, where any conflict about auditing or data update can be fairly arbitrated.

ARCHITECTURE

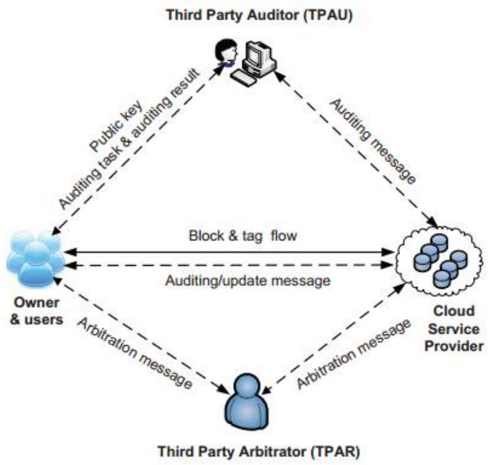

Fig 1 System Architecture.

V. IMPLEMENTATION

Client Module

- This module includes the User registration and user login details.
- Every User need to register while accessing to the cloud.
- Every User will be activated by the Cloud.
- After Cloud activated, every Client need to provide public key to login the user home.
- Public key will be provided by third party auditor.
- Client can view file details and can insert, modify and delete the file with help of TPAR.
- Client will have the TPAR message whenever the user update the file**.**

Third Party Auditor (TPA) Module

- It acts as semi-cloud.
- TPA Provide public key for every user to access the user home page.
- After cloud given auditing proof then only TPA can audit all files.

Third Party Arbitrator (TPA) Module

- It acts as fair dispute for users and cloud.
- Intimate the files message, each time user insert, modify, delete files to cloud.
- Send TPAR message to user and cloud.

Cloud Module

- Activate data client.
- Cloud sends storage auditing proof for all files to TPA.
- Cloud can view the client downloaded files from cloud.
- Cloud will have the TPAR message whenever the user updates the file.

.

V. CONCLUSION

The purpose of this paper is to present an integrity auditing scheme with public verifiability, efficient data dynamics and fair disputes arbitration. To eradicate the constraint of index usage in tag computation and efficiently support data dynamics, it differentiate between block indices and tag indices, and devise an index switcher to keep block-tag index mapping to avoid tag re-computation caused by block update operations, which incurs limited additional overhead. Meanwhile, since both clients and the CSP potentially may misbehave during auditing and data update, the existing threat model is extended in current research to provide fair arbitration for solving disputes between clients and the CSP, which is of vital significance for the deployment and promotion of auditing schemes in the cloud environment. This is achieved by designing arbitration protocols based on the idea of exchanging metadata signatures upon each update operation.

REFERENCES

[1] Y. Deswarte, J.-J. Quisquater, and A. Sa¨ıdane, “Remote integrity checking,” in Proc. 5th Working Conf. Integrity and Intl Control in Information Systems, 2004, pp. 1–11.

[2] D. L. Gazzoni Filho and P. S. L. M. Barreto, “Demonstrating data possession and uncheatable data transfer.” IACR Cryptology ePrint Archive, Report 2006/150, 2006.

[3] A. Juels and B. S. Kaliski Jr, “Pors: Proofs of retrievability for large files,” in Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), 2007, pp. 584–597.

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), 2007, pp. 598–609.

[5] H. Shacham and B. Waters, “Compact proofs of retrievability,” in Proc. 14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 08), 2008, pp. 90–107.

[6] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,” in Proc. 14th European Conf. Research in Computer Security (ESORICS 08), 2009, pp. 355–370.

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

[8] C. Wang, K. Ren, W. Lou, and J. Li, “Toward publicly auditable secure cloud data storage services,” Network, IEEE, vol. 24, no. 4, pp. 19–24, 2010.

[9] C. Erway, A. K¨upc¸ ¨ u, C. Papamanthou, and R. Tamassia, “Dynamic provable data possession,” in Proc. 16th ACM Conf. Computer and Comm. Security (CCS 09), 2009, pp. 213–222.

[10] 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 Proc. ACM Symp. Applied Computing (SAC 11), 2011, pp. 1550–1557.