Volume 3, Issue 5, May 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Implementation of Effective Third Party Auditing for Data Security in Cloud

Vinaya. V
Student M.Tech, Dept of CSE, VTU
Canara Engineering College, Mangalore, India

Sumathi. P
Asst. professor, Dept of CSE, VTU
Canara Engineering College, Mangalore India

Abstract - Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centres, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of effective TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing.

Keywords: Cloud Computing, Third Party Auditor, Integrity.

I. Introduction

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the "software as a service" (SaaS) computing architecture, are transforming data centres into pools of computing service on a huge scale. 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 profound influence on the security and performance of the overall system. Since the security is not provided in cloud, many companies adopt their unique security structure. Introducing a new and uniform security structure for all types of cloud is the problem we are going to tackle. One of the biggest concerns with cloud data storage is that of data integrity verification at un-trusted servers. For example, the storage service provider, which experiences Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. What is more serious is that for saving money and storage space the service provider might neglect to keep or deliberately delete rarely accessed data files which belong to an ordinary client. Consider 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 verifications without the local copy of data files. As data generation is far outpacing data storage it proves costly for small firms to frequently update their hardware whenever additional data is created. Also maintaining the storages can be a difficult task. Storage outsourcing of data to cloud storage helps such firms by reducing the costs of storage, maintenance and personnel. It can also assure a reliable storage of important data by keeping multiple copies of the data thereby reducing the chance of losing data by hardware failures. Storing of user data in the cloud despite its advantages has many interesting security concerns which need to be extensively investigated for making it a reliable solution to the problem of avoiding local storage of data. In this paper we deal with the problem of implementing a protocol for obtaining a proof of data possession in the cloud sometimes referred to as Proof of retrievability (POR).This problem tries to obtain and verify a proof that the data that is stored by a user at a remote data storage in the cloud (called cloud storage archives or simply archives) is not modified by the archive and thereby the integrity of the data is assured. Such verification systems prevent the cloud storage archives from misrepresenting or modifying the data stored at it without the consent of the data owner by using frequent checks on the storage archives. Such checks must allow the data owner to efficiently, frequently, quickly and securely verify that the cloud archive is not cheating the owner. Cheating, in this context, means that the storage archive might delete some of the data or may modify some of the data. To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third party auditor (TPA) to audit the outsourced data when needed. The TPA, who has expertise and capabilities that users do not, can periodically check the integrity of all the data stored in the cloud on behalf of the users, which provides a much more easier and affordable way for the users to ensure their storage correctness in the cloud. Moreover, in addition to help users to evaluate the risk of their subscribed cloud data services, the audit result from TPA would also be beneficial for the cloud service providers to improve their cloud based service platform, and even serve for independent arbitration purposes. In a word, enabling public auditing services will play an important role for this nascent cloud economy to become fully established; where users will need ways to assess risk and gain trust in the cloud.

II. Literature Survey

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system.

We have to analysis the Cloud Computing Outline Survey:

Cloud Computing

Cloud computing providing unlimited infrastructure to store and execute customer data and program. As customers you do not need to own the infrastructure, they are merely accessing or renting; they can forego capital expenditure and consume resources as a service, paying instead for what they use.

- Benefits of Cloud Computing:
- Minimized Capital expenditure
- Location and Device independence
- Utilization and efficiency improvement
- Very high Scalability
- High Computing power

Security a major Concern:

- Security concerns arising because both customer data and program are residing in Provider Premises.
- Security is always a major concern in Open System Architectures.

Data centre Security?

- Professional Security staff utilizing video surveillance, state of the art intrusion detection systems, and other electronic means.
- When an employee no longer has a business need to access datacenter his privileges to access datacenter should be immediately revoked.
- All physical and electronic access to data centres by employees should be logged and audited routinely.
- Audit tools so that users can easily determine how their data is stored, protected, used, and verify policy enforcement.

Data Location:

- When user uses the cloud, user probably won't know exactly where your data is hosted, what country it will be stored in?
- Data should be stored and processed only in specific jurisdictions as define by user.
- Provider should also make a contractual commitment to obey local privacy requirements on behalf of their customers,
- Data-centered policies that are generated when a user provides personal or sensitive information that travels with that information throughout its lifetime to ensure that the information is used only in accordance with the policy.

Backups of Data:

- Data store in database of provider should be redundantly store in multiple physical locations.
- Data that is generated during running of program on instances is all customer data and therefore provider should not perform backups.
- Control of Administrator on Databases.

Network Security:

- Denial of Service: where servers and networks are brought down by a huge amount of network traffic and users are denied the access to a certain Internet based service.
- QOS Violation: through congestion, delaying or dropping packets, or through resource hacking.
- Man in the Middle Attack: To overcome it always use SSL
- IP Spoofing: Spoofing is the creation of TCP/IP packets using somebody else's IP address.
- Solution: Infrastructure will not permit an instance to send traffic with a source IP or MAC address other than its own.

How secure is encryption Scheme:

- Is it possible for all of my data to be fully encrypted?
- What algorithms are used?
- Who holds, maintains and issues the keys? Problem:
- Encryption accidents can make data totally unusable.
- Encryption can complicate availability Solution
- The cloud provider should provide evidence that encryption schemes were designed and tested by experienced specialists.

III. Existing System And Related Work

Cloud storage moves the user's data to large data centres, which are remotely located, on which user does not have any control. However, this unique feature of the cloud poses many new security challenges which need to be clearly understood and resolved. One of the important concerns that need to be addressed is to assure the customer of the

integrity i.e. correctness of his data in the cloud. One of the biggest concerns with cloud data storage is that of data integrity verification at un-trusted servers. For example, the storage service provider, which experiences Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. What is more serious is that for saving money and storage space the service provider might neglect to keep or deliberately delete rarely accessed data files which belong to an ordinary client. Consider 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 verifications without the local copy of data files.

Drawbacks of existing system

- TPA demands retrieval of user data, here privacy is not preserved
- TPA have to remember which key has been used
- These two schemes good for static data not for dynamic data

Recently, much of growing interest has been pursued in the context of remotely stored data verification [2]– [10], [12]–[15]. Ateniese et al. [2] are the first to consider public auditability in their defined "provable data possession" (PDP) model for ensuring possession of files on un trusted storages. In their scheme, they utilize RSA based homomorphic tags for auditing outsourced data, thus public auditability is achieved. However, Ateniese et al. do not consider the case of dynamic data storage, and the direct extension of their scheme from static data storage to dynamic case may suffer design and security problems. In their subsequent work [12], Ateniese et al. propose a dynamic version of the prior PDP scheme. However, the system imposes a priori bound on the number of queries and does not support fully dynamic data operations, i.e., it only allows very basic block operations with limited functionality, and block insertions cannot be supported. In [13], Wang et al. consider dynamic data storage in a distributed scenario, and the proposed challenge-response protocol can both determine the data correctness and locate possible errors. Similar to [12], they only consider partial support for dynamic data operation. Juels et al. [3] describe a "proof of retrievability" (PoR) model, where spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on archive service systems. Specifically, some special blocks called "sentinels" are randomly embedded into the data file F for detection purpose, and F is further encrypted to protect the positions of these special blocks. However, like [12], the number of queries a client can perform is also a fixed priori, and the introduction of pre-computed "sentinels" prevents the development of realizing dynamic data updates. In addition, public auditability is not supported in their scheme. Shacham et al. [4] design an improved PoR scheme with full proofs of security in the security model defined in [3]. They use publicly verifiable homomorphic authenticators built from BLS signatures [16], based on which the proofs can be aggregated into a small authenticator value, and public retrievability is achieved. Still, the authors only consider static data files. Erway et al. [14] was the first to explore constructions for dynamic structure to authenticate the tag information of challenged or updated blocks first before the verification procedure. However, the efficiency of their scheme remains unclear.provable data possession. They extend the PDP model in [2] to support provable updates to stored data files using rank-based authenticated skip lists. This scheme is essentially a fully dynamic version of the PDP solution. To support updates, especially for block insertion, they eliminate the index information in the "tag" computation in Ateniese's PDP model [2] and employ authenticated skip list data.

IV. Proposed System

One of the important concerns that need to be addressed is to assure the customer of the integrity i.e. correctness of his data in the cloud. As the data is physically not accessible to the user the cloud should provide a way for the user to check if the integrity of his data is maintained or is compromised. To enable privacy-preserving public auditing for cloud data storage under the a for mentioned model, our protocol design should achieve the following security and performance guarantees. Public auditability to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users. Storage correctness to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact. Privacy-preserving to ensure that the TPA cannot derive users data content from the information collected during the auditing process. Here we provide a scheme which checks data integrity in the cloud which the customer can employ to check the correctness of his data in the cloud. This can be agreed upon by both the cloud and the customer and can be incorporated in the Service level agreement (SLA). It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted.

V. System Architectural Focus

A representative architecture for cloud data storage is illustrated in Fig. 1 Three different network entities can be identified as follows:

• Client: an entity, which 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 the clients' data.

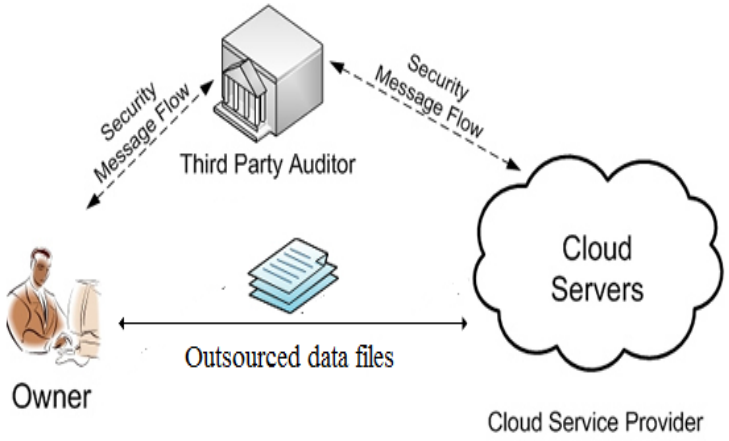

Figure.1 System Architecture

• Third Party Auditor (TPA): an entity, 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 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 those clients do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the monitoring task to a trusted TPA.

A). Integrity Verification

The verifier before storing the file at the archive, preprocesses the file and appends some Meta data to the file and stores at the archive. At the time of verification the verifier uses this Meta data to verify the integrity of the data. It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted. It does not prevent the archive from modifying the data.

Initially client will sends the integrity checking request to the TPA for file f(), TPA will forwards that request to the Server, Server will further fetches the respected file f() from the Server database and generates the Hash code for that file i.e. h(f), and that will be send to the TPA with file name,TPA will generates the signature i.e. SigGen() from the hashcode sent by the Server, TPA will further fetches the old signature from the TPA database i.e. SigAvil(), inally TPA will does the equality check between the SigGen() and SigAvil() Ack will be sent to the Client depend upon the equality checking. This will be shown in the fig. 2.

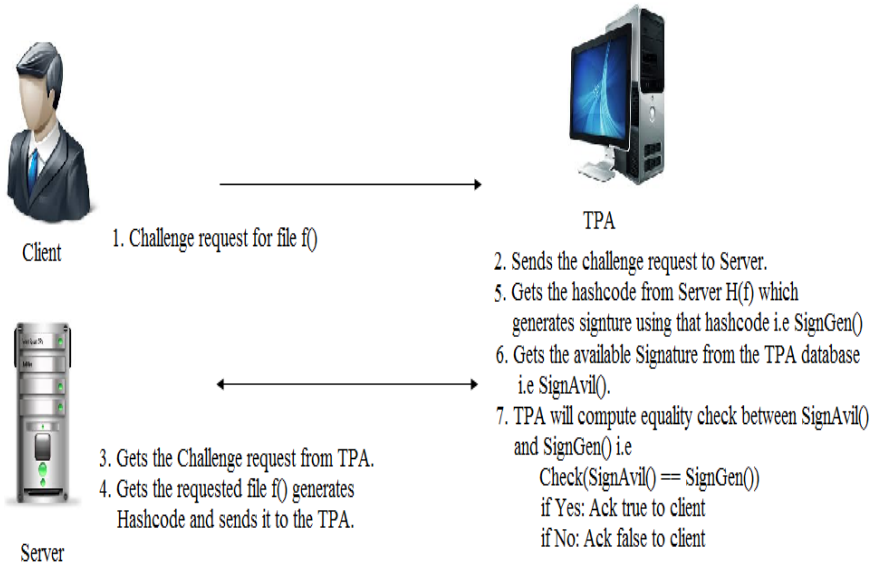

Figure.2 Protocol for Integrity Verification

B). Dynamic Data Operation with Integrity Assurance

Now we show how our scheme can explicitly and efficiently handle fully dynamic data operations like Data Modification including data insertion (I) and data deletion (D) for cloud data storage. Note that in the following descriptions, we assume that the file F and the signature Sig() have already been generated and properly stored at server.

The root metadata R has been signed by the client and stored at the cloud server, so that anyone who has the client's public key can challenge the correctness of data storage.

Initially client sends modify request to Server for file f(), Server will fetches that respected file f() and allows client to modify or update his file, after modification by client Server will update the file and generates the hash code for the updated file and sends it to TPA with file name. TPA will generates Signature for the hash code which is sent by the Server and finally TPA update the Signature in its database for future references. This will be shown in Fig. 3.

Here we provide a scheme which checks data integrity in the cloud which the customer can employ to check the correctness of his data in the cloud. This can be agreed upon by both the cloud and the customer and can be incorporated in the Service level agreement (SLA). It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted.

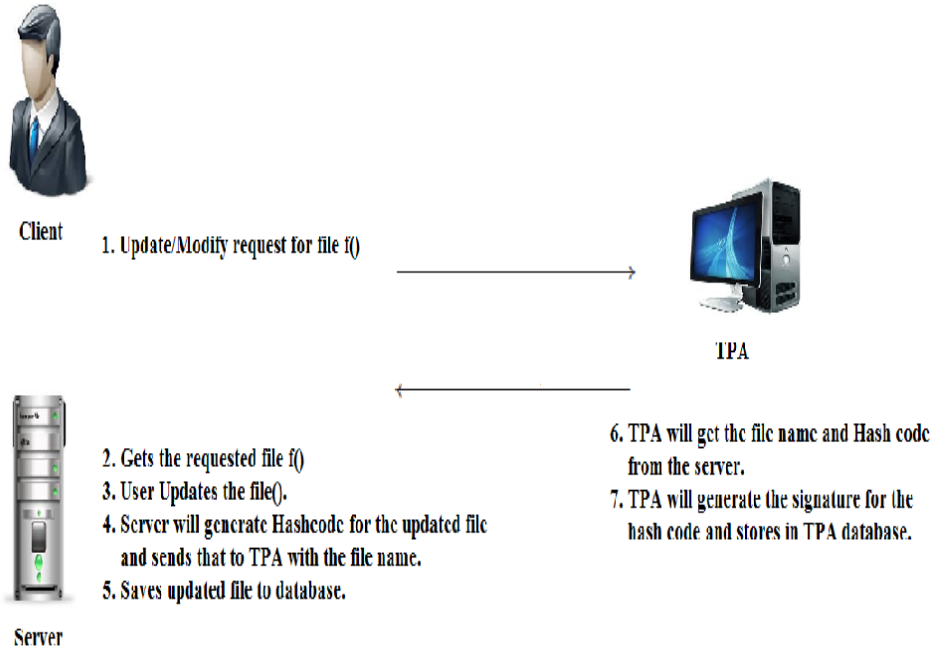

Figure.3 Protocol for provable data update

VI. Conclusion

Here we have presented a data model for secure integrity verification scheme and with data update protocol that dynamic data modification by introducing effective third Party auditor. Here we addressed two issues mainly data correctness and Public auditability which plays very important role in cloud computing features. Public auditability is to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users. Storage correctness to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact. Privacy-preserving to ensure that the TPA cannot derive users data content from the information collected during the auditing process. Here we provide a scheme which checks data integrity in the cloud which the customer can employ to check the correctness of his data in the cloud. This can be agreed upon by both the cloud and the customer and can be incorporated in the Service level agreement (SLA). It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted.

Acknowledgment

First of all I would like to express my heartfelt thanks to Asst. prof. Sumathi. P, for their highly appreciable encouragement and support. Their guidance has been the constant driving force behind my preparation to this Paper. I would also like to thank my lecturers who have been instrumental in inspiring and motivating me with all career guidelines. I am grateful to all the suggestions and hints they have provided with respect to project. Finally, I would thank all my friends who have helped me in collecting the related materials and who have been responsible for improving the quality of this paper by discussing and providing me with the extra information related to the paper. I'm glad to admit that the paper has been a great learning experience and I would certainly look forward to future opportunities like this.

Reference

[1] C. Wang, Q. S.M. Chow, Kui Ren and Qian Wang, "Ensuring data storage security in cloud computing," in December 2011

[2] Amazon.com, "Amazon web services (aws)," Online at http:// aws.amazon.com/, 2009.

[3] Sun Microsystems, Inc., "Building customer trust in cloud computing with transparent security," Online at https://www.sun.com/offers/details/sun transparency.xml, November 2009.

[4] M. Arrington, "Gmail disaster: Reports of mass email deletions," 2006 Online at http://www.techcrunch.com/2006/12/28/gmail-disasterreports-of-mass-aildeletions/December

[5] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors,"at http://www.techcrunch.com/2008/07/10/ mediamaxthelinkup-closes-its-doors/, July 2008.
[6] M. Naor and G. N. Rothblum, "The complexity of online memory checking," in *Proc. of FOCS'05*, Pittsburgh, PA, USA, 2005, pp. 573–584.
[7] E.-C. Chang and J. Xu, "Remote integrity check with dishonest storage server," in *Proc. of ESORICS'08*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 223–237.
[8] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.
[9] A. Oprea, M. K. Reiter, and K. Yang, "Space-efficient block storage integrity," in *Proc. of NDSS'05*, San Diego, CA, USA, 2005.
[10] T. Schwarz and E. L. Miller, "Store, forget, and check: Using algebraic signatures to check remotely administered storage," in *Proc. of ICDCS'06*, Lisboa, Portugal, 2006, pp. 12–12.
[11] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and secure sensor data storage with dynamic integrity assurance," in *Proc. Of IEEE INFOCOM'09*, Rio de Janeiro, Brazil, Appril 2009, pp. 954–962.
[12] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. of SecureComm'08*. New York, NY, USA: ACM, 2008, pp. 1–10.
[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in *Proc. of IWQoS'09*, Charleston, South Carolina, USA, 2009.
[14] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proc. of CCS'09*. Chicago, IL, USA: ACM, 2009.
[15] K. D. Bowers, A. Juels, and A. Oprea, "Hail: A high-availability and integrity layer for cloud storage," in *Proc. of CCS'09*. Chicago, IL, USA: ACM, 2009, pp. 187–198.

AUTHORS PROFILE

Vinaya. V received his B.E Degree in Computer Science & Engineering from Kuvempu University at UBDT College of Engineering Davangere. Karnataka 2009. At present he is pursuing M.Tech Degree in Computer Science & Engineering in Visvesvaraya University Canara Engineering College Mangalore, India.

Sumathi. P received her M.Tech Degree from NITK, Surathkal (VTU), and Currently Pursuing her Ph.D in Web services at VTU Belgaum. She has 13 years of Teaching Experience and Published various International journals and Presented papers in conferences. Presently working as Asst. Prof in Canara Engineering College Mangalore, India.

