Remote Data Integrity Checking in Cloud Computing

Khaba M.V (Assistant Professor/CSE)
RVS School of Engineering ,dindigul
khabarose@gmail.com

M.Santhanalakshmi(Assistant Professor/CSE)
RVS School of Engineering ,dindigul
laksini21@gmail.com

Abstract - Cloud computing is an internet based computing which enables sharing of services. It is very challenging part to keep safely all required data that are needed in many applications for user in cloud. Storing our data in cloud may not be fully trustworthy. Since client doesn't have copy of all stored data, he has to depend on Cloud Service Provider. This work studies the problem of ensuring the integrity and security of data storage in Cloud Computing.This paper, proposes an effective and flexible Batch Audit scheme with dynamic data support to reduce the computation overheads. To ensure the correctness of users data the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the data stored in the cloud.We consider symmetric encryption for effective utilization of outsourced cloud data under the model, it achieve the storage security in multi cloud data storage. The new scheme further supports secure and efficient dynamic operations on data blocks, including data insertion, update, delete and replacement. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colliding attacks.

Index Terms **–Byzantine failure, Batch audit, data dynamics.**

*****

1. INTRODUCTION

Cloud computing is a virtualized resource where we want to store all our data with security measurement so that some application and software can get full benefits using this technology without any local hard disk and server for our data storage. These services are broadly divided into three categories as

1) Infrastructure-as-a-Service. 2) Platform-as-a-Service and 3) Software-as-a-Service [1] [7]. 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 centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers.
One of the biggest concerns with cloud data storage is that of data integrity verification at untrusted 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.

In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models [2]–[11]. In all these works, great efforts are made to design solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data, etc. Considering the role of the verifier in the model, all the schemes presented before fall into two categories: private auditability and public auditability.
Although schemes with private auditability can achieve higher scheme efficiency, public auditability allows anyone, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no private information. Then, clients are able to delegate the evaluation of the service performance to an independent third party auditor (TPA), without devotion of their computation resources. In the cloud, the clients themselves are unreliable or may not be able to afford the overhead of performing frequent integrity checks. Thus, for practical use, it seems more rational to equip the verification protocol with public auditability, which is expected to play a more important role in achieving Economies of scale for Cloud Computing.

Another major concern among previous designs is that of supporting dynamic data operation for cloud data Storage applications. In Cloud Computing, the remotely Stored electronic data might not only be accessed but also Updated by the clients, *e.g.*, through block modification, Deletion and insertion, etc. Unfortunately, the context of remote data storage mainly focus on static data files and the importance of this dynamic data updates has received limited attention so far

[2]–[5]. Moreover, as will be shown later, the direct extension of the current provable data

Possession (PDP) [2] or proof of retrievability (PoR)[2], [3], [4] Schemes to support data dynamics may lead to security loopholes. Although there are many difficulties faced by researchers, it is well believed that supporting dynamic data operation can be of vital importance to the practical application of storage outsourcing services. In view of the key role of batch auditability and data dynamics for cloud data storage, we propose an efficient construction for the seamless integration of these two components in the protocol design. Our contribution can be summarized as follows:

1) We motivate the batch auditing system of data Storage security in Cloud Computing, and then propose a protocol supporting for fully dynamic data operation.

2) We extend our scheme to support scalable and efficient to achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.

2. PROBLEM STATEMENT

2.1System Model

Representative network 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 computation resource to maintain the clients data.

• *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 client's 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 client does not necessarily have the time, feasibility or resources to monitor their data, they can delegate the monitoring task to a trusted TPA. In this paper, we only consider verification schemes with batch auditability: TPA in possession of the public key can act as a verifier. We assume that TPA is unbiased while the server is untrusted. For application purposes, the clients may interact with the cloud servers via CSP to access or retrieve their pre-stored data. More importantly, in practical scenarios, the client may frequently perform block-level operations on the data files.

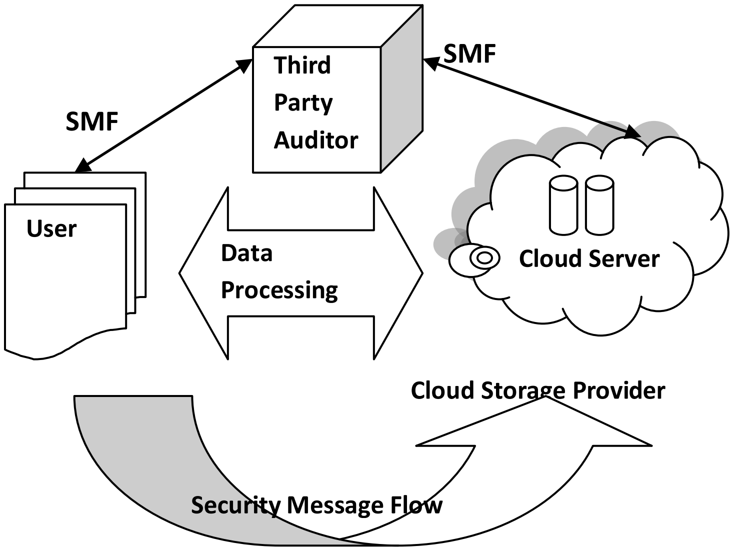

Fig 1: Cloud Data Storage architecture

2.2 Design Methodologies

Our design goals can be summarized as the following:

1) *Batch auditability for storage correctness assurance:* Symmetric Encryption for Effective Utilization of outsourced under the system design model to achieve the security in data storage.

2) *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. The design should be as efficient as possible so as to ensure the seamless integration of batch auditability and dynamic data operation support.

3) *Efficiency*: above goals should be achieved with minimum communication and computation overhead.

3. THE PROPOSED MODEL

In this section, we present our security protocols for cloud data storage service with the aforementioned research goals in mind. We start with some basic solutions aiming to provide integrity assurance of the cloud data and discuss their demerits. Then we present our protocol which supports public auditability and data dynamics. We also show how to extent our main scheme to support batch auditing for TPA upon delegations from multi-user in multi-cloud environment. when we mention the cloud computing issue, enormous threats are raised .One of the major threats are data privacy and integrity. A lot of research discuss this problem and introduce many solutions to decrease the threat of the data privacy and integrity. By their solution, TPA can verify the integrity of files stored by a remote server without knowing any of the file contents. It has three phases: initialization, verification,

3.1Advantages of Proposed Model

By processing the integrity of data using data reading protocol and data management algorithm after and before the entering of data into the cloud, user can assure that all data in cloud must be in protected condition for its trustworthiness. So easily the actual size of stored data before and after in cloud is maintained even though the user himself has done any modification, deletion, and update for his own purpose by using proposed scheme. These processes are carefully done using our proposed scheme. So here user takes full control and process on the data stored in cloud apart from TPA and we can give strong assurance and protection to the data stored in multiple cloud server environment. To avoid server failure and any unexpected error we should put one server restore point in cloud server database for efficient data back up or restore using multi server data comparison method. It is major advantage of our proposed system. This process is done with the help of CSP for cloud database process since we have physical data possession in cloud server.

3.2 Notations and Preliminaries

Bilinear Map: A bilinear map is a map e: G×G → GT, where G is a Gap Diffie-Hellman (GDH) group and GT is another multiplicative cyclic group of prime order p with the following properties [13]: (i) Computable: there exists an efficiently computable algorithm for computing e; (ii) Bilinear: for all h1, h2 ∈ G and a, b ∈ Zp, e(ha1 , hb 2) =e(h1, h2)ab; (iii) Non-degenerate: e(g, g) 6= 1, where g is a generator of G.

Merkle Hash Tree: A Merkle Hash Tree (MHT) is a well studied authentication structure [15], which is intended to efficiently and securely prove that a set of elements are undamaged and unaltered. It is constructed as a binary tree where the leaves in the MHT are the hashes of authentic data values. It depicts an example of authentication. The verifier with the authentic hr requests for {x2, x7} and requires the authentication of the received blocks. The prover provides the verifier with the auxiliary authentication information (AAI) 2 =< h(x1), hd > and 7 =< h(x8), he >. The verifier can then verify x2 and x7 by first computing h(x2), h(x7), hc =h(h(x1)||h(x2))), hf = h(h(x7)||h(x8))), ha = h(hc||hd), hb = h(he||hf) and hr = h(ha||hb), and then checking if the calculated hr is the same as the authentic one.MHT is commonly used to authenticate the values of data blocks. However, in this paper we further employ MHT to authenticate both the values and the positions of data blocks. and extraction for authentication we use the SHA Encryption to maintain the local copy of data.

We treat the leaf nodes as the left-to-right sequence, so any leaf node can be uniquely determined by following this sequence and the way of computing the root in MHT.

4. IMPLEMENTATION

To implement our design, we need to achieve some goals in our model by allowing the TPA to verify the correctness over the cloud data. Additionally, we need to ensure that the cloud server manipulate or alter the user data in the cloud. In our construction, we consider on going the DDP model [7] that is suitable for verifying over the untrusted servers who stores a client’s data. Furthermore, the model is achieved using the digital signature technique.

The digital signature works by taking the user data first, then perform a hash function over it using Secure Hash Algorithm (SHA). After that, computes the signature for the generated hash value by encrypting it with the private key. In the other side, the decryption is done by the public key but the result will be a hash value, and the hash value is not reversible to its original data.

There are three procedures in our model to satisfy the integrity concept:

1. Digital signature part will be done by the user.

2. The CS verifies over the user data in the cloud to check over the manipulation or intrusions in the cloud data.

3. The TPA verifies over the cloud server part to check if the cloud server was manipulating in the user data or not.

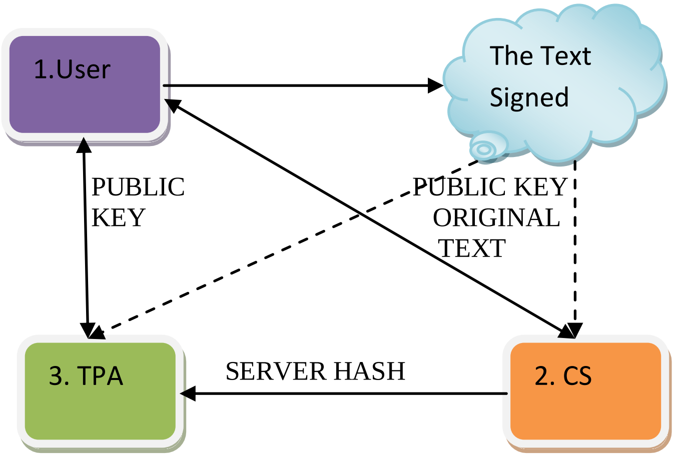

Fig 2: The Proposed Architecture

In next the paragraphs explanation for each entity function in the proposed model:

1)User: User first chose a random parameter to construct the public and the private keys then he\she will sign the data using the private key to be uploaded to the cloud, then he\she send the signed data to the cloud server and deletes its local copy.

2)CS: CS will compute a hash value from the original data to send it to the TPA, and then takes this hash value along with the data signed in the cloud for verification using the public key.

3) TPA: After the cloud server finishes its role, the TPA will be initiated to verify over the cloud server work by taking the hash value from the cloud server. TPA will take the data signed form the cloud and decrypt it with the public key.

5. DATA DYNAMIC PROCESS

To ensure the assurance of the data we can do the operations such as append, deletion, and update. These are the dynamic data operations to be done in the cloud area by user.

5. 1 Append Operation

We may assume that there is any size of GB space allotted by CSP user‘s requirement for any application purpose. Then first, this size is calculated and compared using our technique. It is clearly mentioned in our algorithm specification. In the comparison the storage cloud area is confirmed that it does not have any data in its position for strong integrity. Also the operations such as add, change, and removals by client are processed by space measurement scheme for effective identification of data integrity in cloud database. So if there is any such modification by attack then client can give assurance to the data integrity by successfully following. It is very efficient method in our proposed design compared to any other such type.

5.2 Deletion Operation

First, we want to compare value from existing cloud server. Then, this deletion operation depends on user‘s attempt on his data stored in cloud server using his login operation. The following operation is performed in this deletion operation. If there is number of servers for data selection while deleting the data stored the particular storage server is considered for this. The required data can be deleted using above mentioned algorithm.

5.3 Update Operation

This operation is finally finished after completion of needed action by user. And for update, the above derived algorithms are taken into the account. For new data update, each data block should be updated automatically from already existing value to new updated value. So if we consider this data block as one array formation then, the result is as S= (Si ± $D_{i)}$ depending on the user‘s operation on data. It may be positive or negative depending on the update operation to be performed.

6. BATCHAUDIT ANDTRUSTWORTHINESS

While considering our paper, we have not considered TPA greatly for data integrity purpose since this task almost is completed by user himself. Only the limited authority is given the TPA so that he can‘t be able to modify the contents of user‘s data in strong protection. So he can‘t operate on his own desire for outsourced and in-sourced data in cloud.

7. TYPES OF ATTACKS THROUGH INTERNET LAYER

This attack is based on the web server and network configuration in relevant attacks. For this attack, we can use some method of reverse sweep by coming to the same source from destination for its correctness. Here protocols such as HTTP, SOAP and FTP protocols [2] are considered for clearance. Also this attack is based on the router. Normally the data packet transformation is prevented by these attacks.

8. BENEFITS OF DATA ESTIMATION FROM CLOUD SERVER

While considering all other cloud concept from different authors, here we have proposed new scheme, storage management algorithm with time management for the data calculation to maintain integrity of data. It is one efficient and different method to protect from major data modification, deletion, and append. The main purpose of our proposed scheme is that the total capacity of our allocated space after and before the date update in cloud server is accurately measured by our defined algorithm for the purpose of checking how much of data we already had before any server failure or any internal and external attack. We can maintain integrity of data in well secure manner than other defined methods where the existing systems have.

9. CONCLUSION

To ensure cloud data storage security, it is critical to enable a third party auditor (TPA) to evaluate the service quality from an objective and independent perspective. batch auditability also allows clients to delegate the integrity verification tasks to TPA while they themselves can be unreliable or not be able to commit necessary computation resources performing continuous verifications. Another major concern is how to construct verification protocols that can accommodate *dynamic* data files. In this paper, we explored the problem of providing simultaneous public auditability and data dynamics for remote data integrity check in Cloud Computing. Our construction is deliberately designed to meet these two important goals while efficiency being kept closely in mind. To achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure. From this protection for cloud data, user can be strong belief for his uploaded data for any future purpose or his any other related process without worry

10. FUTURE ENHANCEMENT

Here we leave more ways as Future enhancement to process for maintaining security and integrity of data using read and write protocol for data calculation from cloud data storage in days to come so that user can identify inserting the attempt of different data having same weight in un-trusted cloud server. This read and protect protocol is efficient to identify any data modification into the server with accurate reading and protecting capacity automatically when such attempt is made by known one. In our future study we also have planned to implement the locking protocol in cloud data storage with the help of CSP for data update. It can give clear security to user‘s own data when users complete his requirement.

REFERENCES

[1] Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing Qian Wang1, Cong Wang1, Jin Li1, Kui Ren1, and Wenjing Lou2-Springer-Verlag Berlin Heidelberg -2009.

[2] Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks, Ashley Chonka, YangXiang n, WanleiZhou, AlessioBonti, and Elsevier.

[3] A Multiple-Replica Remote Data Possession Checking Protocol with Public Verifiability-2010 Second International Symposium on Data, Privacy, and E-Commerce.E. H. Miller, A note on reflector arrays (Periodical style—Accepted for publication), *IEEE Trans. Antennas Propagat.*, to be published.

[4] Amazon.com, ―Amazon s3 availability event: July 20, 2008,‖ Onlineathttp://status.aws.amazon.com/s320080720.html, July 2008.

[5] Sun Microsystems, Inc., ―Building customer trust in cloud computing with transparent security, Online at https://www.sun.com/offers/details/suntranspar.xml, November 2009. M. Young, *The Technical Writers Handbook.* Mill Valley, CA: University Science, 1989.

[6] Addressing cloud computing security issues, Future generationcomputersystems (2011)www.elsevier.com/locate/fgcs.

[7] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, ―Provable data possession at untrusted stores, in Proc. of CCS‘07, Alexandria, VA, October 2007, pp. 598–609.

[8] Towards Secure and Dependable Storage Services in Cloud Computing Cong Wang, Student Member, IEEE, Qian Wang, Student Member, IEEE, Kui Ren, Member, IEEE, Ning Cao, Student Member, IEEE, and Wenjing Lou, Senior Member, IEEE- 2011

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," *Proc. of ICDCS'06*, Lisboa, Portugal, 2006, pp. 12–12.

[11] Q. Wang, K. Ran, 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, April 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. Ran, 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.