Authorize Client Inspection and Data Dynamics for Secure Storage in Cloud Computing

P Ram Mohan Rao [#1], P Anjusha [*2], N Subba Reddy [#3] , S Narasimha Rao [#4]
[#1] *Associate Professor& Department of CSE & MLR Institute of Technology Hyderabad, India*

Abstract *Cloud computing is future generation architecture of IT enterprise. We can move the application software and database to centralized data centers, the maintenance and management is easy and not fully dependable. While working with cloud computing we face new security challenges, which have not well understood. This will study the issue of protecting the integrity of data storage in cloud computing. The introduction of a third party auditor (TPA), user side will helpful to verify the dynamic data stored in the cloud. Using TPA we might face issues like block data modification, insertion and deletion without notice. To overcome these kinds of issues we are proposing the public auditability or data dynamics, this achieves both. Initially we identify the challenges and security issues, and then we show how to construct the verification scheme for the integration of the above mentioned two features.*

Keywords— *Authorize Client Inspection, TPA, Data Dynamics, Cloud Computing, Storage Security, Data Integrity.*

I. INTRODUCTION

Various companies are opening up the span of Cloud Computing, with help of web based development using computer technology. These services provided by lower price and more powerful processors, together with the "software as a service" (SaaS) architecture, are replaced data centres into pools of computing service on a huge scale. Now a days we have huge bandwidth for network connectivity will help users to access the data and software which is available in the remote machines.

This new service platform for web, the new data storage architecture in cloud brings new changelings in design, security and performance of overall system. The main issue with cloud data storage is data probity verification at unsafe servers. For example the service provider, the continuous failures occurring occasionally, he might decide to hide the data errors from clients. Other issues might be like for saving money and space chances of deletion of rarely accessed files. Sometimes large amount of data might loss, in that case we get integrity issues, without local copy of data how can we compare and get the details. We can perform integrity operation on existing data.

To resolve the above mentioned issues many people propose the many schemes and methods [2]-[11]. These schemes provided the different solution for different requirement to retrieve the data.

All these methods are divided in to 2 categories private and public inspection. Private auditability can achieve higher scheme efficiency. Public auditability permits anyone, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no hidden information. The users taking help of an independent Third Party Auditor (TPA) to get good performance of services. Individual users might not capable to maintain the regular integrity checking in storage. For real time use to provide the verification protocol with public auditability, which plays major role to get the economies of scale for cloud computing.

Outsourced persons could not involve in the verification process of data stored in cloud. Another main issue as per previous designs is data dynamic operations on different type of cloud applications. The remotely loaded data might not only accessible sometime it will also update like data block modification, insertion and deletion. This will create major issue for all other users , like they might not get original data. Unfortunately, the state-of the-art in the context of remote data storage mostly basis on static data files and the importance of this dynamic data updates has received little attention so far [2]–[5], [7]. Moreover, as will be shown later, the direct extension of the present provable data possession (PDP) [2] or proof of retrievability (POR) [3], [4] schemes to support data dynamics may lead to security ambiguities. Although there are many difficulties faced by researchers, it is well believed that supporting dynamic data operation can be of essential importance to the practical application of storage outsourcing services. In view of the main role of public 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 below:

a) We support the authorize client inspection system of data storage security in Cloud Computing, and propose a methodology supporting for fully dynamic data operations, especially to support block insertion, which is missing in most existing schemes;

b) We extend our scheme to support scalable and efficient public auditing in Cloud Computing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.
c) We prove the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons with the state-of-the-art.

Architecture design:

From below fig:1 we can understand that cloud computing stores data in centralized storage called cloud storage services. Here different types of users can access cloud services like mobile users, laptop users, and computer users. TPA is working as mediator between users and cloud services. TPA will helpful for checking of data stored on cloud computing up on request of users. TPA will access the cloud services and update the data for different application purposes. TPA users able to verity the integrity of shared data up on request of users, without downloading into their local machine.

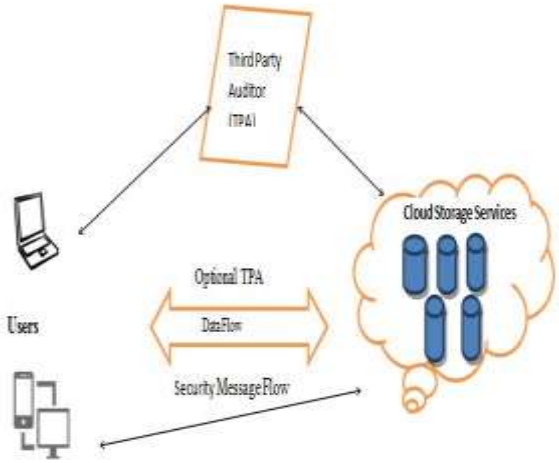


Fig1: Cloud Computing Architecture.

II. Related Work

Majority of the work in storage security and cloud computing is concerned with the integrity of the data at the remote server.
1) Rely on RSA based hash function to verify the file stored at the remote server. With their scheme, the client can perform various challenges with the same metadata. Disadvantage: The inabilities of this scheme exist in the computational difficulty at the server that must exponentiate all the blocks in the file. Miller and Schwarz in
2) Proposed a technique for ensuring the data stored remotely across multiple sites. Algebraic signature was used for this purpose. This scheme makes use of a function to fingerprint the file block and verifies whether the signature of the parity block is matching with the signature of block.
Disadvantages:
1) The calculation difficulty at the server and the client side happens at the cost of linear combination of file blocks.
2) Also, the security of this scheme remains unclear. Public Auditing was first considered by Ateniese et al.
3) The scheme utilizes RSA based homomorphy tags for auditing outsourced data thus achieving public auditing. In this protocol, it is considered that clients need to verify that the server has retained file data without having the server access the entire file and retrieving the data from the server. The model generates probabilistic proofs of obtain by checkout random sets of blocks from the server, that drastically decrease I/O costs. The Provable Data Possession [PDP] model for remote data corroborating supports huge data sets in broadly-distributed storage systems. It is provably-secure scheme for remote data validating. Disadvantages: 1) This method imposes, on client, an overhead of generating metadata. 2) Does not support Dynamic Auditing. 3) Requires more than 1kilo-byte of data for an one time checking. 4) It makes use of only two-party auditing Protocol, which is not efficient as neither the client nor the cloud service provider can give oath to provide balance auditing.
Juels and Kalisiki [4] propose a scheme called "Proofs of Retrievability" (POR) which focuses on static archival of large data files. It makes use of spot checking and error correcting codes to ensure data possession and retrievability. Some special blocks called as "sentinels" are arbitrarily embedded into the file F for detection and then the encryption of the file is carried out in order to protect the position of these sentinel blocks. Unlike PDP scheme the POR scheme cannot be used for public databases. In other words, POR scheme can only be used for confidential data. Disadvantages are: 1) Number of queries clients used is fixed priori. 2) Introduction of sentinel nodes prevents dynamic updating. 3) Each file need to be pre-processed prior to storage at the server. 4) The scheme can only be used for confidential data and not for public databases. 5) Public Auditability is not supported.
Scalable and Efficient Provable Data Possession (S-PDP and E-PDP) protocols contribute to the work of Ateniese et al. [5]. In this paper, a dynamic version of prior PDP scheme depends only on efficient symmetric-key operations in both setup and verification phases[12]. It provides better performance on client side, requires much less storage space and uses less bandwidth (size of challenges and responses is very small, less than a single data block). This scheme is more efficient than POR as it requires no bulk encryption of outsourced data. Disadvantages: 1) The system

imposes a priori bound on the number of queries which can be answered. 2) This concept is applicable only for static data blocks and not dynamic data operations, i.e., it only allow basic block operations with limited functionality. 3) Block insertions cannot be supported and so it is a partially dynamic scheme not fully dynamic. 4) Since the scheme is based on symmetric key cryptography, it is unsuitable for public verification.

The scheme proposed by C.Erway el at [6] is a dynamic auditing protocol that can support the dynamic operations of the data on the cloud servers. This scheme requires the server to send the linear combination of data blocks to the auditor to auditor for verification. This method makes use of Third Party Auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in cloud[9]. It also supports data dynamics via the most general forms of data operation, such as block modification, insertion and deletion [8].

Disadvantages: 1) The main disadvantage of this scheme is that this scheme may leak the data content to the auditor because it requires the server to send linear combinations of data blocks to the auditor for verification. 2) The efficiency of this scheme is not clear.

From the view of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably produces new challenging security threats for number of reasons.

a. Firstly, traditional cryptographic primitives for the cause of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Accordingly, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering different kinds of data for each user stored in the cloud and the demand of long term continuous oath of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging.

b. Furthermore, Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be continually updated by the users, including insertion, deletion, modification, appending, reordering, etc. To protect storage correctness under dynamic data update is hence of paramount importance.

However, these techniques can be useful to ensure the storage correctness without contain users possessing data, cannot address all the security threats in cloud data storage, since they are all focusing on single server scenario and majority of them do not consider dynamic data operations.

As a complementary approach, researchers have also suggested distributed protocols for ensuring storage correctness across multiple servers or peers. Again, none of these distributed schemes is knowledgeable of dynamic data operations. As a result, their applicability in cloud data storage can be drastically restricted.

III. PROPOSED METHOD

Here, we present our security protocols for cloud data storage service with the aforementioned re-search targets in mind. We start with some fundamental solutions aiming to provide integrity assurance of the cloud data and discuss their demerits. Then we present our protocol that supports public auditability and data dynamics. We also show how to extent our main scheme to help batch auditing for TPA upon delegations from multi-users.

Within this paper, we suggested an effective and flexible distributed scheme with definite dynamic data support to guarantee the correctness of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to allow redundancies and guarantee the data dependability. This construction drastically decreases the communication and storage overhead as collated to the traditional replication-based file distribution techniques. By using the homographic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance, also data error localization: whenever data corruption has been detected during the storage correctness confirmation, our scheme can nearly guarantee the simultaneous localization of data errors, i.e., the recognition of the misbehaving server(s).

1. Collated to many of its predecessors, that only provide binary results about the storage condition across the distributed servers, the challenge-response protocol in our work further gives the localization of data error.

2. Unlike most preliminary works for ensuring remote data integrity, the new scheme supports secure and systematic dynamic operations on data blocks, including: update, delete and append.

3. Extensive security and performance review shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data moderation attack, and even server colluding attacks.

Process Flow:

Key Generation:

Choose multiple data, store the data into form of merkle-hash tree structure, count the number of files.

Assigning Keys:

Map the keys to files. Encrypt the files using that corresponding keys. Store the keys and data in a hash table. Because accessing the data using index is less complexity. We cannot do search the whole data, just we search index of the data. So the process will be very speed.

Data Storage in Cloud Computing:
Store that encrypted files in a different location in a cloud server.The requester only having that corresponding keys.The requester gives that keys to the Third Party Auditor.Then the TPA will use that keys and checks the data verification But the TPA cannot see the original data . Only checks the validation using Signature scheme in cryptography.

Integrity Verification:
Decrypt the each and every file in a cloud server.Combine all the files.Check the data size and the size will be same of original data.If any data loss occur for technical problem in a particular file, then put the corresponding encrypt file in that location.We cannot loss security, because we store all the file in a encrypt format.

Data Dynamic:
We are doing some operations in cloud server during run time, data modification, insertion, and deletion.

Batch Auditing:
In a cloud server , lot of users store their files.So each user validate their data using batch system.For that purpose, we use some scheduling algorithms and priority algorithms for avoiding technical problems(i.e) bottleneck, deadlock.So the auditing time will be very less.

IV.RESULTS

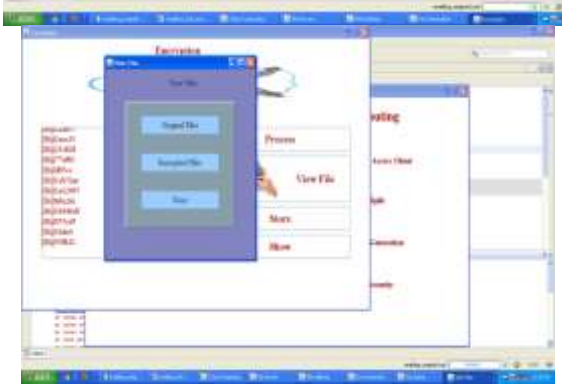
Welcome page:

File Access Client:

File Split:

Key Generation

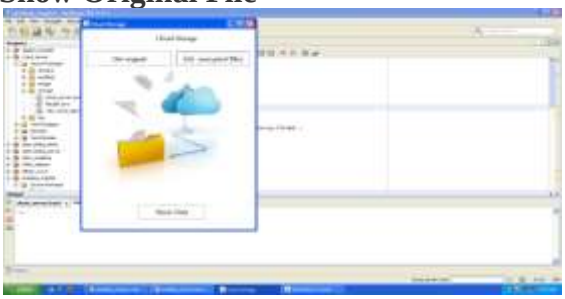
File Security:



Send Original File

User side click send encrypted file

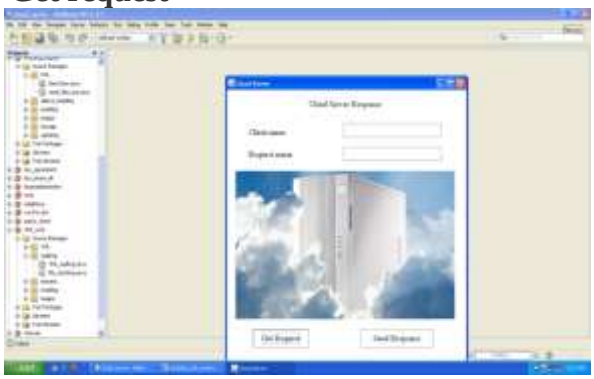
Show Original File



Send Original File

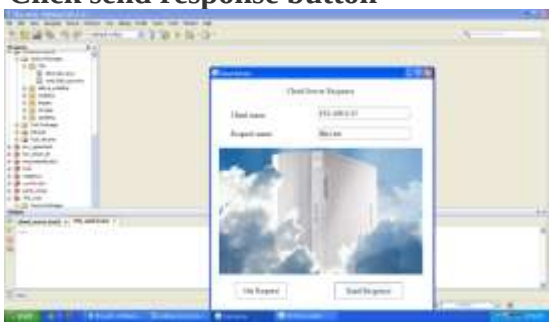
Check stored data by IP

Get request



Click send button

Click send response button



Validate file

Enabling original file

V. CONCLUSIONS

The Our developed system ensures online stored data integrity with help of both public auditability and dynamic data operations. Considering TPA may concurrently handle multiple audit sessions from different user's for their outsourced data files as well as support for data dynamic operation such as block modification, insertion, deletion, verification and taking log history of client performed operation. Also the acknowledgement is getting to cloud space user from the server by the TPA.To achieve efficient data dynamics, we improve the existing evidence of storage models by handling the classic Merkle Hash Tree (MHT) construction for block tag authentication. To support efficient operating of many auditing tasks, we further explore the technique of bilinear aggregate signature to spread our main result into a multi-user setting, where TPA can perform multiple auditing tasks at once

This system can be enhanced in a lot of ways. A backup and recovery system can be added in order to recover the lost or corrupted files from the backup section. During recovery process, instead of fetching entire file from the backup data base, recovery can be done by fetching only the infected block. This will greatly reduce the communication cost. Secondly, a dynamic auditing method can be implemented so that the auditor can periodically check for the files without waiting for the request from the client. This method will completely remove the client's overhead. The client will simply get a notification if any of his files are lost or corrupted and asked for the recovery option. Also instead, the auditor can simply correct the content and maintain the client's data safely. Thirdly, the system can be designed to support multiple auditors so that if an auditor temporarily goes down, the other one can provide his service to the client without delay.

Acknowledgment

I would also like to show my gratitude to the HOD, Department of CSE and Principal, MLR Institute of Technology, Hyderabad for sharing their pearls of wisdom with us during the course of this research, and we thank reviewers for their so-called insights. Any errors are our own and should not tarnish the reputations of these esteemed persons.

REFERENCES

[1] 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. of ESORICS'09*. Saint Malo, France: Springer- Verlag, 2009, pp. 355–370.

[2] 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*. New York, NY, USA: ACM, 2007, pp. 598–609.

[3] A. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in *Proc. of CS'07*. New York, NY, USA: ACM, 2007, pp. 584–597.

[4] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. of ASIACRYPT'08*. Melbourne, Australia: Springer-Verlag, 2008, pp. 90–107.

[5] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," Cryptology ePrint Archive, Report 2008/175, 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.