SECURE DEPENDABLE DATA STORAGE IN CLOUD COMPUTING

[1]R.S.ArchanaVishveswari, [2]P.Selvi
[1]Research Scholar, Department of Information Technology, Jayam College of Engineering and Technology, Dharmapuri
[2]HOD, Department of Information Technology, Jayam College of Engineering and Technology, Dharmapuri
[1]archanars20@gmail.com, [2]selviperuma@gmail.com

Abstract: Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear such a service is also relinquishing user's physical possession of their outsourced data. Which inevitably poses new security risks towards the correctness of the data in cloud? In order to address this new problem and further achieve a secure and dependable cloud storage service. A flexible distributed storage integrity auditing mechanism utilizing the homomorphism token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee but also simultaneously achieves fast data error localization the identification of misbehaving server. Considering the cloud data are dynamic in nature the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification deletion and append. Analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure malicious data modification attack and even server colluding attacks.

Keywords-Data Integrity, dependable distributed storage, error localization, data dynamics, cloud computing.

1. INTRODUCTION

The Cloud Computing Architecture of a cloud solution is the structure of the system, which comprise on - premise and cloud resources, Services, middleware, and software components, geo - location, the externally visible properties of those, and the relationships between them. The term also refers to documentation of a system's cloud computing architecture. Documenting facilitates communication between stakeholders, documents early decisions about high-level design, and allows reuse of design components and patterns between projects.

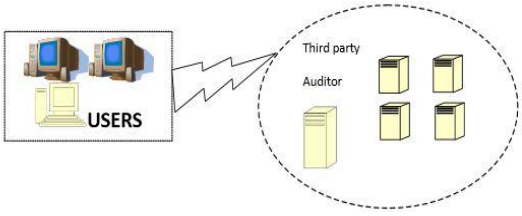


Figure 1: Cloud Architecture

2. PROBLEM DEFINITION:

The clients concern about data security, data integrity, and sharing data with specific band of men and women must be addressed. You can find multiple means of achieving this, example encrypting data on client machine and then storing the information to cloud storage server, computing hash of the information on client machine and storing hash of data in client machine, client trying out the responsibility of sharing the trick key about encryption with specific band of people. Therefore it becomes more tedious for client to keep these information and share such information, more over in the event the device which stores such information is lost or stolen it pose a threat to the total data. Another way could be same storage cloud provider providing the service for secured sharing, hashing, encryption/decryption, but since administrative can have use of both services for maintenance, the security service provided by the cloud storage provider, the information might be compromised. The aforementioned approaches burdens the client by which makes it additionally accountable for securing it data before storing it to the cloud storage

2.1PROBLEM STATEMENT

2.1.1 System Model

Representative network architecture for cloud storage service architecture is illustrated in Fig 1.

Three different network entities can be identified as follows:

- **User:** an entity, who has data to be stored in the cloud and relies on the cloud for data storage and

computation, can be either enterprise or individual customers.

- **Cloud Server (CS):** an entity, which is managed by cloud service provider(CSP) to provide data storage service and has significant storage space and computation resources.
- **Third-Party auditor:** an optional TPA, who has expertise and capabilities that user may not have, is trusted to assess and expose risk of cloud storage services on behalf of the users upon request.

In this model, assume that the point – point communication channels between each cloud server and the user is authenticated an reliable, which can be achieved in practice with little overhead. These authentication handshakes are omitted in the following presentation.

2.1.2 Design Goals

To ensure the security and dependability for cloud data storage under the aforementioned adversary model, we aim to design efficient mechanisms for dynamic data verification and operation and achieve the following goals:

- **Storage correctness:** to ensure users that their data are indeed stored appropriately and kept intact all the time in the cloud.
- **Fast localization of data error:** to effectively locate the malfunctioning server when data corruption has been detected.
- **Dynamic data support:** to maintain the same level of storage correctness assurance even if users modify, delete, or append their data files in the cloud.
- **Dependability:** to enhance data availability against Byzantine failures, malicious data modification and server colluding attacks, i.e., minimizing the effect brought by data errors or server failures.
- **Lightweight:** to enable users to perform storage correctness checks with minimum overhead.

2.1.3 Ensuring cloud data storage

In cloud data storage system, users store their data in the cloud and no longer possess the data locally. Thus, the correctness and availability of the data files being stored on the distributed cloud servers must be guaranteed. One of the key issues is to effectively detect any unauthorized data modification and corruption, possibly due to server compromise and/or random Byzantine failures.

Besides, in the distributed case when such inconsistencies are successfully detected, to find which server the data error lies in is also of great significance, since it can always be the first step to fast recover the storage errors and/or identifying potential threats of external attacks.

2.2 File Retrieval and Error Recovery

Since our layout of file matrix is systematic, the user can reconstruct the original file by downloading the data vectors from the first m servers, assuming that they return the correct response values. Notice that our verification scheme is based on random spot-checking, so the storage correctness assurance is a probabilistic one. However, by choosing system parameters (e.g., r, l, t) appropriately and conducting enough times of verification, we can guarantee the successful file retrieval with high probability. On the other hand, whenever the data corruption is detected, the comparison of pre-computed tokens and received response values can guarantee the identification of misbehaving server(s) (again with high probability), which will be discussed shortly. Therefore, the user can always ask servers to send back blocks of the r rows specified in the challenge and regenerate the correct blocks by erasure correction, shown in Algorithm 3, as long as the number of identified misbehaving servers is less than k. (otherwise, there is no way to recover the corrupted blocks due to lack of redundancy, even if we know the position of misbehaving servers.) The newly recovered blocks can then be redistributed to the misbehaving servers to maintain the correctness of storage.

2.3 Providing dynamic data operation Support

So far, we assumed that **F** represents static or archived data. This model may fit some application scenarios, such as libraries and scientific datasets. However, in cloud data storage, there are many potential scenarios where data stored in the cloud is dynamic, like electronic documents, photos, or log files etc. Therefore, it is crucial to consider the dynamic case, where a user may wish to perform various block-level operations of update, delete and append to modify the data file while maintaining the storage correctness assurance. Since data do not reside at users' local site but at cloud service provider's address domain,

supporting dynamic data operation can be quite challenging. On the one hand, CSP needs to process the data dynamics request without knowing the secret keying material.

On the other hand, users need to ensure that all the dynamic data operation request has been faithfully processed by CSP. To address this problem, we briefly explain our approach methodology here and provide the details later. For any data dynamic operation, the user must first generate the corresponding resulted file blocks and parities. This part of operation has to be carried out by the user, since only he knows the secret matrix P. Besides, to ensure the changes of data blocks correctly reflected in the cloud address domain, the user also needs to modify the corresponding storage verification tokens to accommodate the changes on data blocks.

Only with the accordingly changed storage verification tokens, the previously discussed challenge-response protocol can be carried on successfully even after data dynamics. In other words, these verification tokens help ensure that CSP would correctly execute the processing of any dynamic data operation request. Otherwise, CSP would be caught cheating with high probability in the protocol execution later on. Given this design methodology, the straightforward and trivial way to support these operations is for user to download all the data from the cloud servers and re-compute the whole parity blocks as well as verification tokens. This would clearly be highly inefficient.

In this section, we will show how our scheme can explicitly and efficiently handle dynamic data operations for cloud data storage, by utilizing the linear property of Reed-Solomon code and verification token construction.

3. Third Party Auditor

3.1 Why Third Party Auditor

Third Party Auditor is kind of inspector. There are two categories: private audit ability and public audit ability. Although private audit ability can achieve higher scheme efficiency, public audit ability allows anyone, not just the client (data owner), to challenge the cloud server for the correctness of data storage while keeping no private information. To let off the burden of management of data of the data owner, TPA will audit the data of client. It eliminates the involvement of the client by auditing that whether his data stored in the cloud are indeed intact, which can be important in achieving economies of scale for Cloud Computing. The released audit report would help owners to evaluate the risk of their subscribed cloud data services, and it will also be beneficial to the cloud service provider to improve their cloud based service platform .Hence TPA will help data owner to make sure that his data are safe in the cloud and management of data will be easy and less burdening to data owner

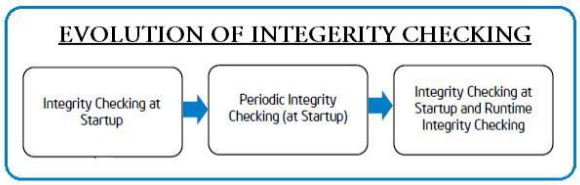
In cloud computing, cloud security is one of the most challenging tasks. Cloud computing entrusts services with users data, software and computation on a published application programming interface over a network. The cloud provides a platform for many types of services. It has a considerable overlap with software as a service (SaaS). End users access cloud based applications through a web browser or a light weight desktop or a mobile app while the business software and data are stored on servers at a remote location. Cloud application providers strive to give the same or better service and performance than if the software programs were installed.

When we talk about cloud Security, maintaining data integrity is one of the most important and difficult task. When we talk about cloud users, they are using cloud services provided by the cloud provider. Again, in case of maintaining the integrity of the data, we cannot trust the service provider to handle the data, as he himself can modify the original data and the integrity may be lost. If a smart hacker hacks the cloud server and steals the data and modifies it then in some cases this modification is not even identified by the cloud provider . So, in this case ,we take the help of a trusted third party auditor to check for the integrity of our data. This third party auditor takes care of our data and makes sure that the data integrity is maintained.

We view the procedure of integrity checking as a key's proficiency within the software, platform, and infrastructure security focus area of our cloud architecture our vision for helping assure ongoing system integrity in a virtualized environment includes an evolution of integrity-checking competences. Each phase in this evolution provides an increasing level of assurance and relies on secure startup enabled.

This evolution begins with one-time integrity checks at system or hypervisor startup, progresses to more frequent periodic integrity checks, and terminates in runtime integrity checks. In a traditional computing environment, increasing restart frequency is difficult because applications are tied to physical servers;

restarting a production server can result in unacceptable application downtime.



Data integrity can be assured by quite a lot of ways. Online integrity checks help to identify and in some cases make good progress from integrity violations. Some systems, instead of performing checks for Integrity, employ preventive methods to reduce the likelihood of an integrity violation. In this section we classify integrity assurance mechanisms into three main types, based on their goals:

- Those that perform defensive steps so as to avoid exact types of integrity damages;
- Those that perform integrity checks and detect integrity violations;
- Those that is skilled of improving from loss once a violation is detected. The check summing techniques helps in detecting integrity violations.

They generally cannot help recovery for two reasons. First, a mismatch between the stored value and the computed value of the checksums just means that one of them was modified, but it does not provide information about which of them is legitimate. Stored checksums are also likely to be modified or corrupted. Second, checksums are generally computed using a one-way hash function and the data cannot be reconstructed given a checksum value.

In proposed method we use RSA algorithm for encryption and decryption which follows the process of digital signature for the message authentication. In our protocol there are three main participants. As discussed above (i) Third Party Auditor (TPA) (ii) User (iii) Cloud Provider. As the initial requirement the user and the TPA generates their own private key and public key with respect to the strong RSA algorithm. The public keys have been shared between them as the part of SLA or in some other ways. Then with respect to the protocol the message is encrypted as well as signed in a unique way.

4. TPABASE SECURITY SCHEME

In the figure below we prepared a model in which Client, CSP and TPA are shown. The client asks the CSP to provide service where CSP authenticate the client and provide a virtual machine by means of Software as a service. In this Vitual Machine (VM), RSA algorithm are used where client encrypt and decrypt the file. In this VM, SHA512 algorithms also there which make the message digest and check the integrity of data. Assurance of their stored data even without the existence of local copies. In case those users do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the tasks to an optional trusted TPA of their respective choices. In our model, we assume that the point-to-point communication channels between each cloud server and the user is authenticated and reliable, which can be achieved in practice with little overhead. Note that we don't address the issue of data privacy in this paper, as in Cloud Computing, data privacy is orthogonal to the problem we study here.

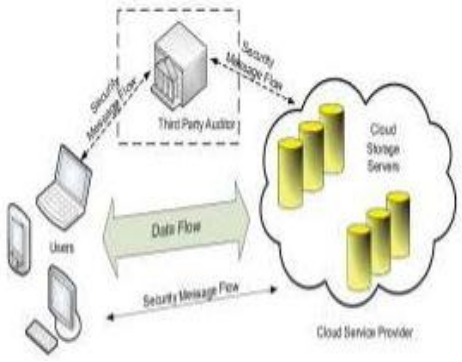


Figure 2: Cloud computing using TPA

4.1 User Level Cryptography

After performing file operation it'll send the information to CSP and TPA. This CSP and TPA can keep our information not solely safe but additionally offer integrity but how it does not make sure that we are going to full trust on TPA. He will send data's of information owner to unauthorized user. If we remove the TPA even it will not solve the matter as a result of CSP may also send the information to unauthorized user and also data owner doesn't get a bonus of TPA. Therefore cryptography is needed at user level. In this scheme encoding and decipherment is completed with

the assistance of RSA formula, colluding together to hide a data loss or corruption incident.

4.2 Mechanism for Data Check Integrity

As data owners no longer physically possess the storage of their data, cryptographic primitives for the purpose of data security protection cannot be directly adopted. In particular, simply downloading the file for its integrity verification is not a practical solution due to the high cost of input/output (I/O) and transmission cost across the network. Also it is not easy to check the data thoroughly and compare with our data. Even the loss of data and recovery of data is also not easy. Considering the large size of the outsourced data and the owner's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for data owners. Hence, to fully ensure data security and save data owners' computation resources, we propose to enable publicly auditable cloud storage services, where data owners can resort to an external third party auditor (TPA) to verify the outsourced data when needed.

5. CONCLUSION

Cloud computing security issues have brought us with great opportunities and challenges. Security in cloud computing can be addressed with TPA and without TPA. In the cloud computing by using the TPA mechanism we can increase the data security which is essentially a distributed storage system. To ensure each data access incontrol and reduce the complexity of cloud computing by help of Advance Encryption Technique (AES). Cryptographic techniques are used to provide secure communication between the client and the cloud. The system ensures that the client's data is stored only on trusted storage servers and it cannot be transferred by malicious system administrators to some corrupt node. Symmetric key sharing is handled with public key cryptography, to achieve faster performance and low computational overhead. The system achieves confidentiality and integrity of the client's data stored in the cloud. Also secure and efficient data dynamic operations such as update delete and append on the data blocks stored in the cloud. Our future goal is to design a secure cloud storage system with TPA which addresses the issues mentioned.

REFERENCES

[1] Cong Wang and KuiRen, Wenjing Lou, Jin Li, Toward Publicly Auditable Secure Cloud Data Storage Services in IEEE Network July/August 2010

[2] N. Gohring, "Amazon's S3 down for several hours," Onlineathttp://www.pcworld.com/businesscenter/article/42549/amazons s3down for several hours.html, 2008.

[3] A. Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. of CCS '07, pp. 584–597, 2007.

[4] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. of Asiacrypt '08, Dec. 2008.

[5] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Cryptology ePrint Archive, Report 2008/175, 2008, http://eprint.iacr.org/.

[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. Of CCS '07, pp. 598–609, 2007.

[7] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. of Secure Comm '08, pp. 1–10, 2008.

[8] T. S. J. Schwarz andE. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc. of ICDCS '06, pp. 12–12, 2006.

[9] M. Lillibridge, S. Elnikety, A. Birrell, M.Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," Proc. of the 2003 USENIXAnnual Technical Conference (General Track), pp. 29–41, 2003.

[10] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage,"Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/