A Secure Auditing and Deduplicating Data in Cloud

R.Sowmiya[1], G.Bala Murugan[2], N. Balasubramanian[3]

[1]Student, Department of Master of Computer Application, Mohamed Sathak Engineering College, Ramanathapuram, India.

[2,3]Assistant Professor, Department of Master of Computer Application, Mohamed Sathak Engineering College, Ramanathapuram, India.

Abstract: **Outsourcing data to cloud service for storage becomes an important trend, which benefits in sparing efforts on heavy data maintenance and management. The outsourced cloud storage is not fully trust worthy; it raises security concerns on how to realize data deduplication in cloud while getting integrity auditing. In this paper, we study the problem of integrity auditing and secure deduplication on cloud data. Specifically, aiming at getting both data integrity and deduplication in cloud, we present two secure systems, namely SecCloud and SecCloud+. SecCloud introduces an auditing entity with maintenance of a MapReduce cloud, which helps clients create data tags before uploading as well as audit the integrity of data having been saved in cloud. Compared with previous work, the computation by user in SecCloud is greatly reduced during the file uploading and auditing phases. SecCloud+ is designed motivated by the fact that customers always want to encrypt their data before uploading, and enables integrity auditing and secure deduplication on encrypted data.**

Keywords: **Cloud Storage, Data De-Duplicating and secure auditing.**

1.INTRODUCTION

Cloud computing provides a simple way to access servers, storage, databases and set of application services over the Internet. Cloud computing helps save considerable capital costs without spending on in-house server storage and application needs.A domain, in the context of networking, refers to any group of users, workstations, devices, computers and databaseservers that share different types of data through network resources. A domain cloud computing is also used to assign specific resource privileges, such as user accounts. Cloud is a model for penetrating user data's, on demand network services to access the secure data. Cloud computing has an three classes.

1.IaaS (infrastructure as a service) provides access to computation resources as per user basics.

2. SaaS (software as a service) is a simple application; it is delivered to thousands of users from the resource pool.

3.PaaS (platform as a service) uses the building blocks of the vendor's deployment environment.

If the user need to access the data from shared pool, administrator confirms user is an authorized person to take the Cloud data storage services includes some of the entities.(i)Administrator controls the user details, file insertion, file access, file deletion and the time of user presents in the network to access the cloud data's. (ii)TPA checks the correctness of cloud data and also Some performances are used to establish the auditing concepts.(iii) Users access the cloud data as per

demand services. Users recover more useful information from multiple repositories and no limitation to access the particular storage part in the shared pool.

While cloud storage system has been extensivelyadopted,it fails to accommodate some emerging needs such as the abilities of auditing integrity of cloud files by cloud clients and detecting duplicated files by cloud servers. We demonstrate both problems. The first problem is integrity auditing. The cloud server is able to relieve clients from the heavy burden of storage for management and maintenance. The most difference of cloud storage from traditional in-house storage is that the data is transferred through Internet and stored in an uncertain domain, not under control of the clients at all, which unavoidably raises clients great anxieties on the integrity of their data. The second problem is secure deduplication. The fast adoption of cloud services is escorted by increasing volumes of data stored at remote cloud servers. These attacks originate from the reason that the client owns a given file (or block of data) is only based on a static and short value.

2.BACKGROUND & RELATED WORKS:

Since our work is related to both integrity auditing and secure deduplication data in cloud.

1.Integrity Auditing

The provable data possession (PDP) is for assuring that the cl oud servers possess

the target files without retrieving or downloading the whole data. Fundamentally, PDP is a probabilistic proof protocol by sampling a random set of blocks and asking the servers to prove that they exactly possess these blocks, and the verifier only upholding a small amount of metadata is able to perform the integrity checking.

Another line of work supporting integrity auditing is proof of retrievability. Compared with PDP, POR not just promises the cloud servers possess the target files, but also contracts their full recovery. Inclients apply erasure codes and make authenticators for each block for verifiability and retrievability

2.Secure Deduplication

Deduplication is a technique where the server stores only a single copy of each file, irrespective on how many clients asked to store that file,the disk space of cloud servers as network bandwidth are saved. Though, trivial client-side deduplication leads to the leakage of side channel information. In order to restrict the leakage of side channel information, introduced the proof of ownership protocol which lets a client efficiently prove to a server that that the client exactly holds this file. Several proof of ownership protocols based on the hash tree are proposed to enable secure client-side deduplication.

3. PERFORMANCE ANALYSIS

In this section, we will provide a experimental evaluation of our proposed schemes. We build our test bed by using 64-bit t2. Micro Linux servers in Amazon EC2 platform as the auditing server and storage server. In order to achieve λ = 80-bit security, the prime order p of thebilinear group G and GT are respectively chosen as 160 and 512 bits in length. We also set the block size as 4 KB and each block includes 25 sectors

1.Tag generation

Tag generation shows the time cost of slave node in MapReduce for generating file tags. It is clear the time cost of slave node is growing with the size of file. This is because the more blocks in file, the more homomorphic signatures are needed to be computed by slave node for file uploading. We also need to notice that there does not exist much computational load difference between common slave nodes and the reducer. Compared with the common slave nodes, reducer only additionally involves in a number of multiplications, which is lightweight operation. Noting that, the procedure of tag generation could be handled in pre-processing,

and it is not needed for client to wait until uploading file.

2. File Auditing

Before examine the time cost of file auditing, we need to firstly make analysis and identify the number of challengingblocks in our integrity auditing protocol. According, if ρ fraction of the file is corrupted, through asking the proof of a constant m blocks of this file, the verifier can detect the misbehaviourwith probability $\alpha = 1 - (1 - \rho)^m$. To capture the spirit of probabilistic auditing, we set the probability confidence α = 70%; 85% and 99%, and draw the relationships between ρ and m.It demonstrates that if we want to achieve low (i.e., 70%), medium (i.e., 85%) and high (i.e., 99%) confidence of detecting any small fraction of corruption, we have to respectively ask for 130, 190 and 460 blocks for challenge. Now, we come back evaluate the time cost of file auditing, which shows the time cost of auditing for detecting the misbehaviour of cloud storage respectively with 70%, 85% and 99% confidence. Obviously, as the growth of the number of blocks for challenge, the time cost for response from cloud storage server is increasing. Congruently, the time cost at auditor grows with the number of challenge blocks as well.

4. SECCLOUD SYSTEM MODEL

Directing at allowing for auditable and deduplicated storage, we propose the SecCloud system. In the SecCloud system, we have three entities:

1.Cloud Clients:

Cloud Clients have large data files to be stored and rely on the cloud for data maintenance and computation. They can be either individual consumers or commercial organizations.

2.***Cloud Servers:***

Cloud Servers virtualize the resources according to the requirements of clients and expose them as storage pools. Typically, the cloud clients may buy or lease storage capacity from cloud servers, and store their individual data in these accepted or rented spaces for future utilization.

3.Auditor:

Auditor which helps clients upload and audit their outsourced data maintains a MapReduce cloud and acts like a certificate authority. This statement presumes that the auditor is associated with a pair of public and private keys. Its public key is made available to the other entities in the system.

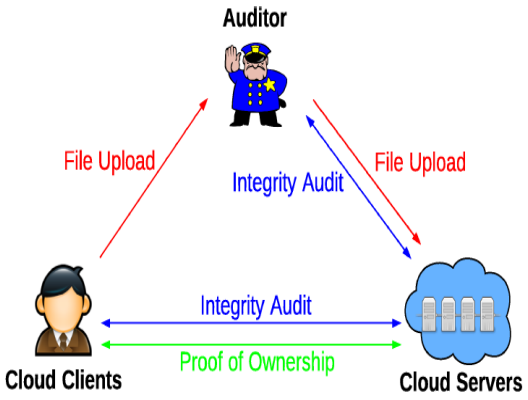

Figure 1:system model

The SecCloud system supporting file-level deduplication includes the following three protocols respectively highlighted by colour in the figure 1.

1.File Uploading Protocol:

This protocol aims at allowing clients to upload files via the auditor. Specifically, the file uploading protocol includes three methods:

(cloud client → cloud server): Client takes the duplicate check with the cloud server to confirm if such a file is stored in cloud storage or not before uploading a file. If there is a duplicate, another protocol called Proof of Ownership will be run between the client and the cloud storage server.

(cloud client → auditor): Client uploads files to the auditor, and receives a receipt from auditor.

(auditor → cloud server): Auditor helps generate a set of tags for the uploading file, and send them along with this file to cloud server

2.Integrity Auditing Protocol:

It is an interactive protocol for integrity verification and allowed to be initialized by any entity except the cloud server. In this protocol, the cloud serverplays the role of prover, while the auditor or client works as the verifier. This protocol includes two methods:

(cloud client/auditor → cloud server): Verifier (i.e., client or auditor) generates a set of challenges and sends them to the prover (i.e., cloud server)

(cloud server → cloud client/auditor): Based on the stored files and file tags, prover (i.e., cloud server) tries to prove that it exactly owns the target file by sending the proof back to verifier (i.e., cloud client or auditor). At the end of this protocol, verifier outputs true if the integrity verification is passed.

3. Proof of Ownership Protocol:

It is an interactive protocol initialized at the cloud server for verifying that the client exactly owns a claimed file. This protocol is typically triggered along with file uploading protocol to prevent the leakage of side channel information. On the contrast to integrity auditing protocol, in PoW the cloud server works as verifier, although the client plays the role of prover. This protocol also includes two methods.

(cloud server → client): Cloud server generates a set of challenges and sends them to the client.

(client → cloud server): The client responds with the proof for file ownership, and cloud server finally verifies the validity of proof.

i)Integrity Auditing:

The first design goal of this work is to provide the capability of verifying correctness of the remotely stored data. The integrity verification further requires two features those are public verification and stateless verification.

ii)Secure Deduplication:

The second design goal of this work is secure deduplication. In other words, it requires that the cloud server is able to decrease the storage space by keeping only one copy of the same file. Notice that, regarding to secure deduplication, our objective is distinguished from previous work in that we propose a method for allowing both deduplication over files and tags.

iii)Cost-Effective:

The computational overhead for providing integrity auditing and secure deduplication should not show a major additional cost to traditional cloud storage, nor should they alter the way either uploading or downloading operation.

5.CONCLUSION

Working at both data integrity auditing and deduplication in cloud, I exhibit SecCloud and SecCloud+.Seccloud propose an element with use of MapReduce cloud and it helps customer make info labels before transferring and review the integrity of information in cloud. Besides, SecCloud empowers secure deduplication through is presenting a Proof of Ownership convention and integrity and keeping away from the concept of side divert data in information deduplication which have been Contrasted and past work, the calculation by client in SecCloud is extremely transferring and inspecting stages leads an reduce of the document. SecCloud+ is a pushed development influenced by the way that clients dependably need to encode their information before transferring, and takes into process of consideration trustworthiness evaluating and secure deduplication specifically on scrambled information.

6.REFERENCE

1.Jingwei Li, Jin Li, DongqingXie and Zhang Cai, "Secure Auditing and De-duplicating Data in Cloud", IEEE Transactions on Computers Vol: Pp No: 99 Year 2015.

2. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communication of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

3. J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," in IEEE Conference on Communications and Network Security (CNS), 2013, pp. 145–153.

4.S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in Proceedings of the 18th ACM Conference on Computer and Communications Security. ACM, 2011, pp. 491–500.

5. S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Server aided encryption for de-duplicated storage," in Proceedings of the 22Nd USENIX Conference on Security, ser. SEC'13. Washington, D.C.: USENIX Association, 2013, pp. 179–194. [Online].

6. C.Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 213–222.

7.F. Seb´e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. on Knowl. and Data Eng., vol. 20, no. 8, pp. 1034–1038, 2008.

8. E. Stefanov, M. van Dijk, A. Juels, and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks," in Proceedings of the 28th Annual Computer Security Applications Conference, ser. ACSAC '12. New York, NY, , 2012, pp. 229–238.

9.M. Azraoui, K. Elkhiyaoui, R. Molva, and M. O¨ nen, "Stealthguard: Proofs of retrievability with hidden watchdogs," in Computer Security - ESORICS 2014, ser. Lecture Notes in Computer Science, M. Kutyłowski and J. Vaidya, Eds., vol. 8712. Springer International Publishing, 2014, pp. 239–256.

10. J. Li, X. Tan, X. Chen, and D. Wong, "An efficient proof of retrievability with public auditing in cloud computing," in 5th International ConferenNetworking and collaborative systems(INCoS),2013,PP.