Volume 4 Issue III, March 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Deduplicating Data in Cloud and Secure Auditing

Shelke P. Shruti[1], Inamdar A. Muskan[2], Unawane S. Samiksha[3], Shinde P. Shubhahasta[4], Avinash B. Anap [5]
[12345]Computer Department, P.Dr.V.V.P. Polytechnic, Loni

Abstract: ***As the cloud computing technology develops during the last decade, outsourcing data to cloud service for storage becomes an attractive trend, which benefits in sparing efforts on heavy data maintenance and management. Nevertheless, since the outsourced cloud storage is not fully trustworthy, it raises security concerns on how to realize data deduplication in cloud while achieving integrity auditing. In this work, we study the problem of integrity auditing and secure deduplication on cloud data. Specifically, aiming at achieving both data integrity and deduplication in cloud, we propose two secure systems, namelySecCloud and SecCloud+. SecCloud introduces an auditing entity with maintenance of a MapReduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of data having been stored 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, Encryption, Advanced Encryption Standard (AES), Integrity.

I. INTRODUCTION

Cloud computing (The Fifth Generation of Computing) is a term used to describe both a platform and type of application. A cloud computing platform dynamically provisions, configures, reconfigures, and DE provisions servers as needed.

Servers in the cloud can be physical machines or virtual machines. Advanced clouds typically include other computing resources such as storage area networks (SANs), network equipment, firewall and other security devices. Cloud computing also describes applications that are extended to be accessible through the Internet. These cloud applications use large data centers and powerful servers that host Web applications and Web services. Anyone with a suitable Internet connection and a standard browser can access a cloud application. As the cloud computing technology develops during the last decade, outsourcing data to cloud service for storage becomes an attractive trend, which benefits in sparing efforts on heavy data maintenance and management. Nevertheless, since the outsourced cloud storage is not fully trustworthy, it raises security concerns on how to realize data DE duplication in cloud while achieving integrity auditing.

In this work, we study the problem of integrity auditing and secure DE duplication on cloud data. Specifically, aiming at achieving both data integrity and DE duplication in cloud, we propose two secure systems, namely SecCloud and SecCloud+. Sec Cloud introduces an auditing entity with maintenance of a Map Reduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of data having been stored 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 DE duplication on encrypted data.

A cloud is a pool of virtualized computer resources. A cloud can:
Host a variety of different workloads, including batch-style back-end jobs and interactive, user-facing applications
Allow workloads to be deployed and scaled-out quickly through the rapid provisioning of virtual machines or physical machines
Support redundant, self-recovering, highly scalable programming models that allow workloads to recover from many unavoidable hardware/software failures
Monitor resource use in real time to enable rebalancing of allocations when needed
Cloud storage is a model of networked enterprise storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Cloud storage provides customers with benefits, ranging from cost saving and simplified convenience, to mobility opportunities and scalable service. These great features attract more and more customers to utilize and storage their personal data to the cloud storage: according to the analysis report, the volume of data in cloud is expected to achieve 40 trillion gigabytes in 2020. Even though cloud storage system has been widely adopted, it fails to accommodate some important emerging needs such as the abilities of auditing integrity of cloud files by cloud clients and detecting duplicated files by cloud servers. We illustrate both problems below. The first problem is integrity auditing. The cloud server is able to relieve clients from the heavy burden of storage management and maintenance. The most difference of cloud storage from traditional in-house storage is that the data is transferred via Internet and stored in an uncertain domain, not under control of the clients at all, which inevitably raises clients great concerns on the integrity of their data.

SECCLOUD

we describe our proposed SecCloud system. Specifically, we begin with giving the system model of Sec- Cloud as well as introducing the design goals for SecCloud. In what follows, we illustrate the proposed SecCloud in detail.

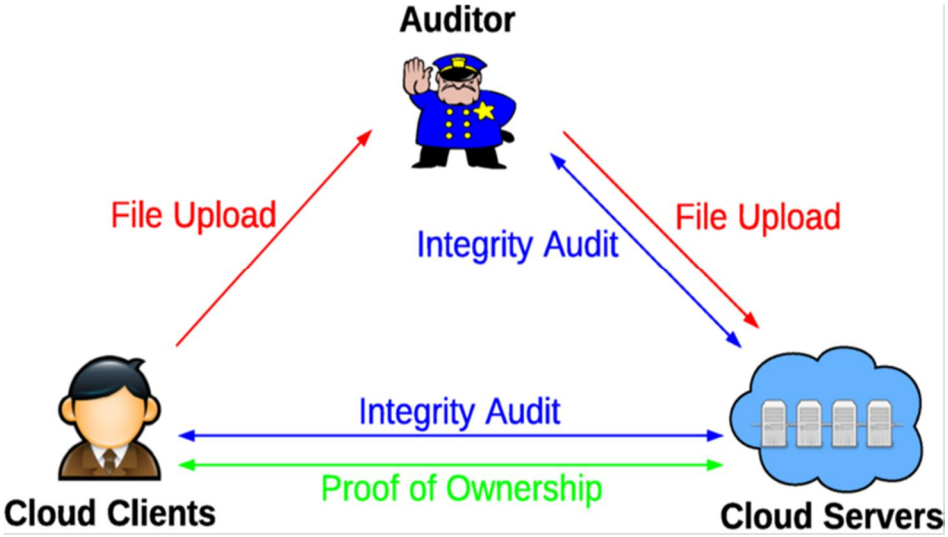

Fig. SECCLOUD Architecture

A. System Model

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

• 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;

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

• Auditor which helps clients upload and audit their outsourced data maintains a MapReduce cloud and acts like a

certificate authority. This assumption 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. The SecCloud system supporting file-level deduplication includes the following three protocols respectively highlighted by red, blue and green in Fig.

1. File Uploading Protocol:

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

• Phase 1 (cloud client → cloud server): client performs 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. Otherwise, the following protocols (including phase 2 and phase 3) are run between these two entities.

• Phase 2 (cloud client → auditor): client uploads files to the auditor, and receives a receipt from auditor.

• Phase 3 (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 server plays the role of prover, while the auditor or client works as the verifier. This protocol includes two phases:

• Phase 1 (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).

• Phase 2 (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, while the client plays the role of prover.

II. LITERATURE SURVEY

We have been gathering information on how a college functions and how can we reach each entity of the college by a certain medium which can be handy for all.So we chose cloud as the medium to reach the students. We chose JAVA as the front end and Cloud as the backend. Cloud storage is a model of networked enterprise storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Cloud storage provides customers with benefits, ranging from cost saving and simplified convenience, to mobility opportunities and scalable service. It has been widely adopted, it fails to accommodate some important emerging needs such as the abilities of auditing integrity of cloud files by cloud clients and detecting duplicated files by cloud servers. We illustrate both problems below. The first problem is integrity auditing. The cloud server is able to relieve clients from the heavy burden of storage management and maintenance.

The most difference of cloud storage from traditional in-house storage is that the data is transferred via Internet and stored in an uncertain domain, not under control of the clients at all, which inevitably raises clients great concerns on the integrity of their data. These concerns originate from the fact that the cloud storage is susceptible to security threats from both outside and inside of the cloud.

III. RELATED WORK

Since our work is related to both integrity auditing and secure deduplication, we review the works in both areas in the following subsections, respectively. A. Integrity Auditing The definition of provable data possession (PDP) was introduced by Ateniese et al. [5][6] for assuring that the cloud servers possess the target files without retrieving or downloading the whole data. Essentially, 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 maintaining a small amount of metadata is able to perform the integrity checking. After Ateniese et

al.'s proposal [5], several works concerned on how to realize PDP on dynamic scenario: Ateniese et al. [7] proposed a dynamic PDP schema but without insertion operation; Erway et al. [8] improved Ateniese et al.'s work [7] and supported insertion by introducing authenticated flip table; A similar work has also been contributed in [9]. Nevertheless, these proposals [5][7][8][9] suffer from the computational overhead for tag generation at the client. To fix this issue, Wang et al. [10] proposed proxy PDP in public clouds. Zhu et al. [11] proposed the cooperative PDP in multi-cloud storage. Another line of work supporting integrity auditing is proof of retrievability (POR) [12]. Compared with PDP, POR not merely assures the cloud servers possess the target files, but also guarantees their full recovery. In [12], clients apply erasure codes and generate authenticators for each block for verifiability and retrievability. In order to achieve efficient data dynamics, Wang et al. [13] improved the POR model by manipulating the classic Merkle hash tree construction for block tag authentication. Xu and Chang [14] proposed to improve the POR schema in [12] with polynomial commitment for reducing communication cost. Stefanov et al. [15] proposed a POR protocol over authenticated file system subject to frequent changes. Azraoui et al. [16] combined the privacy-preserving word search algorithm with the insertion in data segments of randomly generated short bit sequences, and developed a new POR protocol. Li et al. [17] considered a new cloud storage architecture with two independent cloud servers for integrity auditing to reduce the computation load at client side. Recently, Li et al. [18] utilized the key-disperse paradigm to fix the issue of a significant number of convergent keys in convergent encryption. B. Secure Deduplication Deduplication is a technique where the server stores only a single copy of each file, regardless of how many clients asked to store that file, such that the disk space of cloud servers as well as network bandwidth are saved. However, trivial client side deduplication leads to the leakage of side channel information. For example, a server telling a client that it need not send the file reveals that some other client has the exact same file, which could be sensitive information in some case. In order to restrict the leakage of side channel information, Halevi et al. [3] 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 Merkle hash tree are proposed [3] to
enable secure client-side deduplication. Pietro and Sorniotti [19] proposed an efficient proof of ownership scheme by choosing the projection of a file onto some randomly selected bit-positions as the file proof. Another line of work for secure deduplication focuses on the confidentiality of deduplicated data and considers to make deduplication on encrypted data. Ng et al. [20] firstly introduced the private data deduplication as a complement of public data deduplication protocols of Halevi et al. [3]. Convergent encryption [21] is a promising cryptographic primitive for ensuring data privacy in deduplication. Bellare et al. [22]
formalized this primitive as message-locked encryption, and explored its application in space-efficient secure outsourced storage. Abadi et al. [23] further strengthened Bellare et al's security definitions [22] by considering plaintext distributions that may depend on the public parameters of the schemas. Regarding the practical implementation of convergent encryption for securing deduplication, Keelveedhi et al. [4] designed the DupLESS system in which clients encrypt under file-based keys derived from a key server via an oblivious pseudorandom function protocol. As stated before, all the works illustrated above considers either integrity auditing or deduplication, while in this paper, we attempt to solve both problems simultaneously. In addition, it is worthwhile noting that our work is also distinguished with [2] which audits cloud data with deduplication, because
we also consider to 1) outsource the computation of tag generation, 2) audit and deduplicate encrypted data in the proposed protocols.

IV. ALGORITHM USED

The Advanced Encryption Standard (AES), also known asRijndael is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES is based on the Rijndael cipherdeveloped by two Belgiancryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal to NIST during the AES selection process. Rijndael is a family of ciphers with different key and block sizes. For AES, NIST selected three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256 bits. AES has been adopted by the U.S. government and is now used worldwide. It supersedes the Data Encryption Standard (DES),which was published in 1977. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

In the United States, AES was announced by the NIST as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001.This announcement followed a five-year standardization process in which fifteen competing designs were presented and evaluated, before the Rijndael cipher was selected as the most suitable . AES became effective as a federal government standard on May 26, 2002 after approval by the Secretary of Commerce. AES is included in the ISO/IEC 18033-3 standard. AES is available in many

Volume 4 Issue III, March 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

different encryption packages, and is the first publicly accessible and open[vague]cipher approved by the National Security Agency (NSA) for top secretinformation when used in an NSA approved cryptographic module. The name Rijndael is a play on the names of the two inventors. It is also a combination of the Dutch name for the Rhine river and adale.

AES is based on the Rijndael cipherdeveloped by two Belgiancryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal to NIST during the AES selection process. Rijndael is a family of ciphers with different key and block sizes.For AES, NIST selected three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256bits.AES has been adopted by the U.S. government and is now used worldwide. It supersedes the Data Encryption Standard (DES)which was published in 1977. The algorithm described by AES is a symmetric-key algorithm ,meaning the same key is used for both encrypting and decrypting the data.AES became effective as a federal government standard on May 26, 2002 after approval by the Secretary of Commerce. AES is included in the ISO/IEC 18033-3 standard. AES is available in many different encryption packages, and is the first publicly accessible and opencipher approved by the National Security Agency (NSA) for top secretinformation when used in an NSA approved cryptographic module The name Rijndael is a play on the names of the two inventors (Joan Daemen and Vincent Rijmen). It is also a combination of the Dutch name for the Rhine River and ad ale.

V. METHODOLOGY

We can consider an example related with a customer database in a college consisting of student's information along with his id card information. Every college will have much type of confidential data, for maintaining these data, must have database and should expert with DBMS. So SDBAAS will help the manager to store every database into cloud after providing security. The schema for storing such information will be in the form of tables. Some tables containing personal information of the user and some tables containing information regarding to id cards and will be mapped using their ids. This particular information can be stored in a college DB database as follows: Student table (Student Id, StudentName, StudentAddress, Customer StudentPhone, Student rDOB).Membershiptable (StudentId,Password, PasswordQuestion, PasswordAnswer). Student1table (Id, Enrollment no, Admission date). Any user can register into this web based college application. User will register with his personal information. Then system will provide id number for each Student. This will stored in Student table. Then he can login to the application with username and password, perform registration like login, verifying etc. The manager will save every user's id in the table and upload into cloud after encryption with SQL aware encryption scheme or adaptive encryption scheme. Whenever user or manager needs information, provide SQL commands. The SDBAAS provide corresponding result after downloading and decrypting the databases.

VI. FLOWCHART

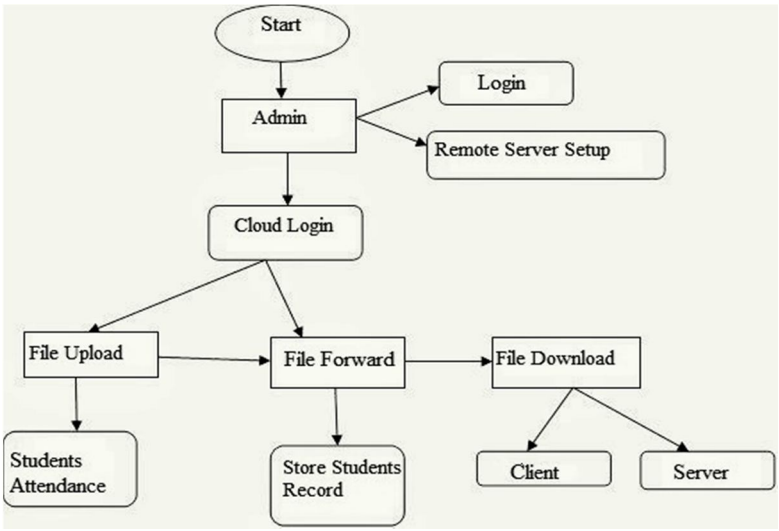

Figure: Flow Chart of Data Base Maintenance

VII. CONCLUSION

Aiming at achieving both data integrity and deduplication in cloud, we propose SecCloud and SecCloud+. SecCloud in- troduces an auditing entity with maintenance of a MapReduce cloud, which helps clients generate data tags before uploading as well as audit the

integrity of data having been stored in cloud. In addition, Sec Cloud enables secure DE duplication through introducing a Proof of Ownership protocol and pre- venting the leakage of side channel information in data DE duplication. Compared with previous work, the computation by user in SecCloud is greatly reduced during the file uploading and auditing phases.

SecCloud+ is an advanced construction motivated by the fact that customers always want to encrypt their data before uploading, and allows for integrity auditing and secure DE duplication directly on encrypted data.

Cloud Computing is the fastest growing part of IT

Tremendous benefits to customers of all sizes

Cloud services are simpler to acquire and scale up or down

Key opportunity for application and infrastructure vendors

Public clouds work great for some but not all applications

Private clouds offer many benefits for internal applications

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] S. Keelveedhi, M. Bellare, and T. Ristenpart, “Dupless: Server- aided encryption for deduplicated storage,” in Proceedings of the 22Nd USENIX Conference on Security, ser. SEC’13. Washington, D.C.: USENIX Association, 2013, pp. 179–194. [Online]. Avail- able: https://www.usenix.org/conference/usenixsecurity13/technical- sessions/presentation/bellare

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in Proceed- ings of the 14th ACM Conference on Computer and Communications Security, ser. CCS ’07. New York, NY, USA: ACM, 2007, pp. 598– 609.

[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, “Remote data checking using provable data possession,” ACM Trans. Inf. Syst. Secur., vol. 14, no. 1, pp. 12:1–12:34, 2011.

[7] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, “Scalable and efficient provable data possession,” in Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, ser. SecureComm ’08. New York, NY, USA: ACM, 2008, pp. 9:1–9:10.

[8] 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.

[9] 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.

[10] H. Wang, “Proxy provable data possession in public clouds,” IEEE

