Scientific Journal of Impact Factor (SJIF): 3.134

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 1, January -2016

Auditing and Avoiding Data Deduplication in Cloud

Supriya Shedge[1], Renuka Patil[2], Dhrupat Patle[3], Prof. Anuja Bugad[4]

[1,2,3,4]*Department of Information Technology, G. S. Moze Engineering College,Pune,*

Abstract *— Cloud storage is a place where customers, users stored data as per their requirements. Cloud storage is not fully secure for data deduplication while achieving integrity auditing. In this work the problem of integrity auditing and secure deduplication on cloud data by using map reduce technique. In our work, SecCloud and SecCloud+ is used to auditing and avoiding deduplication of data. SecCloud introduced on auditing entity with Map Reduce cloud and Reliability. Map Reduce is a framework which helps clients to generate data tags before, also it enhance the Reliability of system because it distribute the data in block and stores on distributed network or distributed file system and master node. For security purpose customer always wants to encrypt data before uploading and enables integrity auditing and secure deduplication on encrypted data with the SecCloud+.IN our work used MAC protocol for tag generation.*

Keywords- authentication, deduplication, integrity auditing cloud storage

I. INTRODUCTION

Despite the fact that cloud stockpiling framework has been generally embraced, it neglects to oblige some critical emerging needs, for example, the capacities of auditing integrity of cloud files by cloud customers and detecting copied files by cloud servers. We show both issues underneath. The first issue is integrity auditing. The cloud server has the capacity alleviate customers from the substantial weight of capacity administration and maintenance. The most distinction of cloud stockpiling from customary in-house stockpiling is that the data is exchanged by means of Internet and put away in an uncertain domain, not under control of the customers by any stretch of the imagination, which inevitably raises customers extraordinary worries on the integrity of their data. These worries originate from the way that the cloud stockpiling is defenseless to security dangers from both outside and inside of the cloud, and the uncontrolled cloud servers might inactively conceal some data misfortune incidents from the customers to maintain their notoriety. In addition genuine is that for saving cash and space, the cloud servers may even effectively and purposely dispose of once in a while got to data files belonging to an ordinary customer. Considering the substantial size of the outsourced data files and the customers' constrained asset abilities, the first issue is summed up as in what manner can the customer efficiently perform periodical integrity verifications even without the neighborhood duplicate of data files.

The cloud storage is powerless to security dangers from both outside and inside of the cloud [1], and the uncontrolled cloud servers might inactively conceal some information misfortune episodes from the customers to keep up their notoriety. Deduplication would prompt various dangers conceivably influencing the stockpiling framework [3][2], for instance, a server telling a customer that it (i.e., the customer) does not require to send the record uncovers that some other customer has the precise same record, which could be touchy sometimes customers dependably need to encrypt their information before transferring, for reasons extending from individual protection to corporate strategy, we bring a key server into SecCloud as with [4] and propose the SecCloud+ pattern.

II. LITERATURE REVIEW

1) PROOFS OF OWNERSHIP IN REMOTE STORAGE SYSTEMS

Author: ShaiHalevi

Cloud storage systems are becoming increasingly popular. A promising technology that keeps their cost down is deduplication, which stores only a single copy of repeating data. Client-side deduplication attempts to identify deduplication opportunities already at the client and save the bandwidth of uploading copies of existing files to the server. In this work we identify attacks that exploit client-side deduplication, allowing an attacker to gain access to arbitrary-size files of other users based on a very small hash signature of these files. More specifically, an attacker who knows the hash signature of a file can convince the storage service that it owns that file, hence the server lets the attacker download the entire file.

2) PROVABLE DATA POSSESSION AT UNTRUSTED STORES

Author: Giuseppe Ateniese

We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs

of possession by sa3mpling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems.

3) SECURE AUDITING AND DEDUPLICATING DATA IN CLOUD

Author: Jingwei Li,Zhang Cai

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, namely SecCloud and SecCloud+.SecCloud introduces an auditing entity with a 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.

4) DupLESS: SERVER-AIDED ENCRYPTION FOR DEDUPLICATED STORAGE

Author: MihirBellare

Cloud storage service providers such as Dropbox, Mozy, and others perform deduplication to save space by only storing one copy of each file uploaded. Should clients conventionally encrypt their files, however, savings are lost. Message-locked encryption (the most prominent manifestation of which is convergent encryption) resolves this tension. However it is inherently subject to brute-force attacks that can recover files falling into a known set. We propose an architecture that provides secure deduplicated storage resisting brute-force attacks, and realize it in a system called DupLESS. In DupLESS, clients encrypt under message-based keys obtained from a key-server via an oblivious PRF protocol. It enables clients to store encrypted data with an existing service, have the service perform deduplication on their behalf, and yet achieves strong confidentiality guarantees. We show that encryption for deduplicated storage can achieve performance and space savings close to that of using the storage service with plaintext data.

5) SECURE AUDITING DATA IN CLOUD

Author: Jingwei Li, Jin Li, Dongqing Xie and Zhang Cai

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. These concerns originate from the fact that the cloud storage is susceptible to security threats from both outside and inside of the cloud , and the uncontrolled cloud servers may passively hide some data loss incidents from the clients to maintain their reputation. What is more serious is that for saving money and space, the cloud servers might even actively and deliberately discard rarely accessed data files belonging to an ordinary client. Considering the large size of the outsourced data files and the clients' constrained resource capabilities, the first problem is generalized as how can the client efficiently perform periodical integrity verifications even without the local copy of data files .The second problem is secure deduplication. The rapid adoption of cloud services is accompanied by increasing volumes of data stored at remote cloud servers. Among these remote stored files, most of them are duplicated: according to a recent survey by EMC , 75% of recent digital data is duplicated copies. This fact raises a technology namely deduplication, in which the cloud servers would like to deduplicate by keeping only a single copy for each file and make a link to the file for every client who owns or asks to store the same file. Unfortunately, this action of deduplication would lead to a number of threats potentially affecting the storage system , for example, a server telling a client that it does not need to send the file reveals that some other client has the exact same file, which could be sensitive sometimes. These attacks originate from the reason that the proof that the client owns a given file is solely based on a

static, short value. Thus, the second problem is generalized as how can the cloud servers efficiently confirm that the client owns the uploaded file before creating a link to this file for him/her. In this paper, aiming at achieving data integrity and deduplication in cloud, we propose two secure systems namely SecCloud and SecCloud+. SecCloud introduces an auditing entity with a maintenance of MapReduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of data having been stored in cloud. This design fixes the issue of previous work that the computational load at user or auditor is too huge for tag generation. For completeness of fine-grained, the functionality of auditing designed in SecCloud is supported on both block level and sector level. In addition, SecCoud also enables secure deduplication. Notice that the "security" considered in SecCoud is the prevention of leakage of side channel information.

III. SURVEY OF PROPOSED SYSTEM

We determine that our proposed SecCloud framework has accomplished both integrity auditing and file deduplication. Be that as it may, it can't keep the cloud servers from knowing the substance of files having been put away. In other words, the functionalities of integrity auditing and secure deduplication are just forced on plain files. In this area, we propose SecCloud+, which takes into account integrity auditing and deduplication on scrambled files. Framework Model Compared with SecCloud, our proposed SecCloud+ involves an extra trusted element, to be specific key server, which is in charge of assigning customers with mystery key (according to the file content) for encrypting files. This construction modeling is in line with the late work. However, our work is distinguished with the past work by allowing for integrity auditing on encoded data. SecCloud+ takes after the same three protocols (i.e., the file uploading protocol, the integrity auditing protocol and the proof of proprietorship protocol) as with SecCloud. The main distinction is the file uploading protocol in SecCloud+ involves an extra stage for correspondence between cloud customer and key server. That is, the customer needs to speak with the key server to get the merged key for encrypting the uploading file before the phase in SeeCloud.

IV. Mathematical Model

Let S be whole system,
S={I,P,O}
Where,
I-Input
P- Procedure,
O- Output,
Now,
I={TG,DD,POW,C,PR,V}
TG= Tag Generation.
DD= Deduplication.
POW= Proof of Ownership.
C= Challenge by TPA to CSP.
PR= Proof by CSP to TPA.
V= Verify By TPA

Procedure(P):

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):
Phase 2 (cloud client → auditor):
Phase 3 (auditor → 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):
Phase 2 (cloud server → cloud client/auditor):
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. This protocol also includes two phases :
Phase 1 (cloud server → client):
Phase 2 (client → cloud server):

Output (O)

Integrity Auditing. The first design goal of this work is to provide the capability of verifying correctness of the remotely stored data.

Secure Deduplication. The second design goal of this work is secure deduplication. In other words, it requires that the cloud server is able to reduce the storage space by keeping only one copy of the same file.

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

Following are steps to be followed:

ALGORITHMS:

KeyGen(F) : The key generation algorithm takes a file content F as input and outputs the convergent key ckF of F;

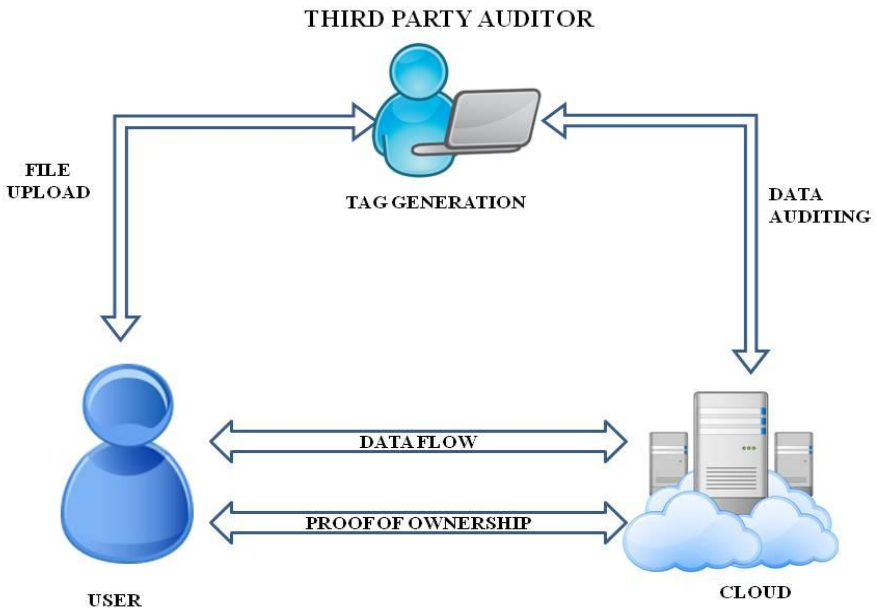
• Encrypt(ckF;F) : The encryption algorithm takes the convergent key ckF and file content F as input and outputs the ciphertext ctF;

• Decrypt(ckF; ctF) : The decryption algorithm takes the convergent key ckF and ciphertext ctF as input and outputs the plain file F;

• TagGen(F) : The tag generation algorithm takes a file content F as input and outputs the tag tagF of F. Notice that in this paper, we also allow TagGen(·) to generate the (same) tag from the corresponding ciphertext as with [6][7].

V. SYSTEM ARCHITECTURE

.



VI. CONCLUSION

Aiming at achieving both data integrity and deduplication in cloud, we propose SecCloud and SecCloud+. SecCloud introduces an auditing substance with maintenance of a MapReduce cloud, which assists customers with generating data labels before uploading and additionally reviews the integrity of data having been put away in cloud. Furthermore, SecCoud empowers secure deduplication through introducing a Proof of Ownership protocol and preventing the leakage of side channel information in data deduplication. Contrasted and past work, the calculation by client in SecCloud is incredibly diminished during the file uploading and auditing stages. SecCloud+ is a propelled development persuaded by the way that clients constantly need to encode their data before uploading, and takes into account integrity auditing and secure deduplication straightforwardly on scrambled data.

ACKNOWLEDGMENT

We might want to thank the analysts and also distributers for making their assets accessible. We additionally appreciative to commentator for their significant recommendations furthermore thank the school powers for giving the obliged base and backing.

VII 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.179194.[Online].Available:https://www.usenix.org/conference/usenixsecurity13/technicalsessions/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] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Advances in Cryptology – EUROCRYPT 2013*, ser. Lecture Notes in Computer Science, T. Johansson and P. Nguyen, Eds. Springer Berlin Heidelberg, 2013, vol. 7881, pp.296–312.

[7] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp.1615–1625, June 2014.

