ISSN 2395-1621

Securely Surveying and Removing Twin Data InCloud

[#1]PritamGote, [#2]SwapnilAmbi, [#3]Dinesh Divekar, [#4]DipakShinde, [#5]Prof. RupaliPatil

[1]pritamgote03@gmail.com
[2]swapnilmbi92@gmail.com
[3]dineshdivekar52@gmail.com
[4]dipak.shinde20@gmail.com

[#12345]ParvatiBaiGenbaMoze College of Engineering

ABSTRACT

As the cloud computing technology develops throughout the last decade, outsourcing information to cloud service for storage becomes a beautiful trend, that edges in frugal efforts on significant information maintenance and management. All the same, since the outsourced cloud storage isn't absolutely trustworthy, it raises security issues on a way to understand information deduplication in cloud while achieving integrity auditing. In this work, we tend to study the matter of integrity auditing and secure deduplication on cloud information. Specifically, aiming at achieving each information integrity and deduplication in cloud, we propose two secure systems, specifically SecCloud and SecCloud+. SecCloud introduces associate degree auditing entity with a maintenance of a MapReduce cloud, That helps purchasers generate information tags before uploading likewise as audit the integrity of knowledge having been keep in cloud. Compared with previous work, the computation by user in SecCloud is greatly reduced throughout the file uploading and auditing phases. SecCloud+ is intended motivated by the actual fact that customers invariably wish to write in code their information before uploading, and permits integrity auditing and secure deduplication on encrypted information.

ARTICLE INFO

Article History

Received 30th March 2016

Received in revised form :

1st March 2016

Accepted : 2nd April 2016

Published online :

4th April 2016

Keywords:-SecCloud, SecCloud++,Deduplication, Encryption, integrity auditing, MapReduce Cloud, Convergent Key.

I. INTRODUCTION

Cloud storage may be a model of networked enterprise storage where information is keep in virtualized pools of storage that area unit generally hosted by third parties. Cloud storage provides customers with advantages, starting from price saving and simplified convenience, to quality opportunities and ascendable service.The first drawback is integrity auditing. The first drawback is generalized as however will the consumer expeditiously perform periodical integrity verifications even while not the native copy of information files. The second drawback is secure deduplication. The second problem is generalized as however will the cloud servers expeditiously confirm that the consumer (with an explicit degree assurance) owns the uploaded file (or block) before making a link to the present file (or block) for him/her. during this paper, aiming at achieving information integrity and deduplicationin cloud, we have a tendency to propose 2 secure systems specifically SecCloud and SecCloud+.SecCloud introduces AN auditing entity with a maintenance of a MapReduce cloud, that helps shoppers generate information tags before uploading additionally as audit the integrity of information having been keep in cloud.Motivated by the very fact that customers continually wish to code their information before uploading, for reasons starting from personal privacy to company policy, we have a tendency to introduce a key server into SecCloudand propose the SecCloud+ schema. Besides supporting integrity auditing and secure deduplication, SecCloud+ permits the guarantee of file confidentiality.

II. RELATED WORKS

1. Integrity Auditing

The definition of obvious knowledge possession (PDP) was introduced by Ateniese et al. [5][6] for reassuring that the cloud servers possess the target files while not retrieving

ordownloading the entire knowledge. primarily, PDP could be a probabilisticproof protocol by sampling a random set of blocks and askingthe servers to prove that they precisely possess these blocks, and the champion solely maintaining atiny low quantity of information is able to perform the integrity checking. when Ateniese et al.'s proposal [5], many works involved on the way to notice PDP on dynamic scenario: Ateniese et al. [7] planned a dynamic PDP schema however while not insertion operation; Erway et al. [8] improved Ateniese et al.'s work [7] and supported insertion by introducing echt flip table; an analogous work has also been contributed in [9]. still, these proposals [5][7][8][9] suffer from the process overhead for tag generation at the consumer. to repair this issue, Wang et al. [10] proposed proxy PDP publically clouds. Zhu et al. [11] planned the cooperative PDP in multi-cloud storage. Another line of labor 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], purchasers apply erasure codes and generate authenticators for every block for verifiability and retrievability. so as to attain economical knowledge dynamics, Wang et al. [13] improved the POR model by manipulating the classic Merkle hash tree construction for block tag authentication. Xu and Chang Jiang [14] planned to boost the POR schema in [12] with polynomial commitment for reducing communication price.Stefanov et al. [15] planned a POR protocol over echt filing system subject to frequent changes. Azraoui et al. [16] combined the privacy-preserving word search algorithmic rule with the insertion in knowledge segments of randomly generated short bit sequences, and developed a brand newPOR protocol. Li et al. [17] thought-about a brand new cloud storage architecture with 2 freelance cloud servers for integrity auditing to scale back the computation load at consumer facet. Recently, Li et al. [18] used the key-disperse paradigm to repair the issue of a big range of focused keys in focused encryption.

1. Secure Deduplication

Deduplication may be a technique wherever the server stores solely a single copy of every file, notwithstanding what number purchasers asked to store that file, specified the space of cloud servers likewise as network information measure ar saved. However, trivial shopper aspect deduplication ends up in the discharge of aspect channel data. as an example, a server telling a shopper that it needn't send the file reveals that another shopper has the exact same file, that can be sensitive data in some case. In order to limit the discharge of aspect channel data, Halevi et al. [3] introduced the proof of possession protocol which lets a shopper with efficiency sway a server that that the shopper specifically holds this file. Many proof of possession protocols supported the Merkle hash tree ar projected [3] to enable secure client-side deduplication. Pietro and Sorniotti[19] projected AN economical proof of possession theme by choosing the projection of a file onto some willy-nilly hand-picked bit-positions because the file proof. Another line of labor for secure deduplication focuses on the confidentiality of deduplicated knowledge and considers to create deduplication on encrypted knowledge. Ng et al. [20] foremost introduced the non-public knowledge deduplication as a complement of public data deduplication protocols of Halevi et al. [3]. Convergent encryption [21] may be a promising cryptological primitive for ensuring knowledge privacy in deduplication. Bellare et al. [22] formalized this primitive as message-locked cryptography, and explored its application in space-efficient secure outsourced storage. Regarding the sensible implementation of focused cryptography for securing deduplication,Keelveedhi et al. [4] designed the DupLESS system within which purchasers encode underneath file-based keys derived from a key server via AN oblivious pseudorandomfunction protocol. As declared before, all the works illustrated on top of considers either integrity auditing or deduplication, whereas during this paper, we decide to solve each issues at the same time. additionally, it is worthy noting that our work is additionally distinguished with [2] that audits cloud knowledge with deduplication, because we conjointly deliberate to 1) source the computation of tag generation, 2) audit and deduplicate encrypted knowledge within the proposed protocols.

2. Convergent Encryption

Convergent encoding provides information confidentiality in deduplication. A user (or information owner) derives a convergent key from the information content and encrypts the information copy with the convergent key. Additionally, the user derives a tag for the information copy, specified the tag are going to be used to discover duplicates. Here, we have a tendency to assume that the tag correctness property holds, i.e., if two information copies are identical, then their tags are identical.

III. PROPOSED SYSTEM

1. SecCloud

In this section, 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.

A] System Model

Targeting at allowing for auditable and deduplicated storage, we propose the SecCloudSystem.It is three entities

1. Cloud Client have large data file to be stored.
2. Cloud Sever virtualize the resource according to beneed.
3. Auditor which assist clients upload the data.

In this model, there are three protocols:

1.File Uploading Protocol: This protocol aims at permitting clients to transfer files via the auditor. Specifically, the file uploading protocol includes 3 phases:

section one (cloud consumer → cloud server): consumer performs the duplicate refer to the cloud server to verify if such a file is hold on in cloud storage or not before uploading a file. If there's a reproduction, another protocol called Proof of possession are going to be run between the consumer and the cloud storage server. Otherwise, the subsequent protocols (including section a pair of and section 3) are run between these 2 entities.

• section two (cloud consumer → auditor): consumer uploads files to the auditor, and receives a receipt fraudster.

• section three (auditor → cloud server): auditor helps generate a set of tags for the uploading file, and send them onwith this file to cloud server.

2.IntegrityAuditing Protocol: it's Associate in Nursing interactive protocol for integrity verification and allowed to be initialized by any entity except the cloud server. during this protocol, the cloud serverplays the role of prover, whereas the auditor or consumer works as the booster. This protocol includes 2 phases:

• section one (cloud client/auditor → cloud server): booster (i.e., consumer or auditor) generates a group of challenges and sends them to the prover (i.e., cloud server).

• section a pair of (cloud server → cloud client/auditor): supported the hold on files and file tags, prover (i.e., cloud server) tries to prove that it precisely owns the target file by sending the proof back to booster (i.e., cloud consumer or auditor). At the tip of this protocol, booster outputs true if the integrity verification is passed.

3.Proof of possession Protocol: it's Associate in nursing interactive protocol initialized at the cloud server for corroborative that the consumer exactly owns a claimed file. This protocol is usually triggered along with file uploading protocol to stop the outflow of side channel data. On the distinction to integrity auditing protocol, in POW the cloud server works as booster, while the client plays the role of prover. This protocol conjointly includes 2 phases

• section one (cloud server → client): cloud server generates a set of challenges and sends them to the consumer.

• section a pair of (client → cloud server): the consumer responds with the proof for file possession, and cloud server finally verifies the validity of proof.

SecCloud Architecture

2. SecCloud++

We specify that our planned SecCloud system has achieved both integrity auditing and file deduplication. However, it cannot forestall the cloud servers from knowing the content of files having been hold on. In different words, the functionalities of integrity auditing and secure deduplication square measure solely obligatory on plain files. During this section, we have a tendency to propose SecCloud+, which allows for integrity auditing and deduplication on encrypted files.

A. System Model

Compared with SecCloud, our planned SecCloud+ involves an additional trusty entity, specifically key server, whichis liable for assignment shoppers with secret key (accordingto the file content) for encrypting files. This design is inline with the recent work.However our work is distinguished with the previous work by allowing integrity auditing on encrypted knowledge. SecCloud+ follows identical 3 protocols (i.e., the file uploading protocol, the integrity auditing protocol and therefore the proof of possession protocol) like SecCloud. The only difference is that the file uploading protocol in SecCloud+ involves an additional part for communication between cloud consumer and key server. That is, the consumer must communicate with the key server to urge the convergent key for encrypting the uploading file before the part a pair of in SecCloud. Unlike SecCloud, another style goals of file confidentiality is desired in SecCloud+ as follows.

• File Confidentiality. The planning goal of file confidentiality requires to stop the cloud servers from accessing the content of files. Specially, we have a tendency to need that the goal of file confidentiality must be immune to “dictionary attack”. That is, even the adversaries have pre-knowledge of the “dictionary” which incorporates all the possible files, they still cannot recover the target file.

IV. CONCLUSION

Aiming at achieving each information integrity and deduplication in cloud, we have a tendency to propose

SecCloud and SecCloud+. SecCloud introduces an auditing entity with maintenance of a MapReduce cloud, that helps shoppers generate information tags before uploading as well as audit the integrity of information having been keep in cloud. additionally, SecCoud permits secure deduplication through introducing a symbol of possession protocol and preventing the outflow of aspect channel data in knowledge deduplication. Compared with previous work, the computation by user in SecCloud is greatly reduced throughout the file uploading and auditing phases. SecCloud+ is a sophisticated construction motivated by the very fact that customers continually need to write their information before uploading, and permits for integrity auditing and secure deduplication directly on encrypted information

ACKNOWLEDGEMENT

Specially thanks to our project guide Mrs.RupaliPatil, ,Mr.ShrikantDhamdhere,Mr.VijayRathi,Mr.Akram AnasariAnd all supporting staff of our project.

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 Communicationsand 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 ACMConference on Computer and Communications Security. ACM, 2011, pp. 491–500.

[4] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Serveraidedencryption for deduplicated storage," in Proceedings of the22Nd USENIX Conference on Security, ser. SEC'13. Washington, D.C.: USENIX Association, 2013, pp. 179–194.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedingsof the 14th ACM Conference on Computer and CommunicationsSecurity, 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 4thInternational Conference on Security and Privacy in CommunicationNetowrks, 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 Conferenceon 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 Transactions on Services Computing, vol. 6, no. 4, pp. 551–559, 2013.

[11] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEE Transactionson Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231– 2244, 2012.

[12] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proceedings of the 14th International Conference on the Theory andApplication of Cryptology and Information Security: Advances in Cryptology,ser. ASIACRYPT '08. Springer Berlin Heidelberg, 2008, pp. 90–107.

[13] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Computer Security – ESORICS 2009, M. Backes and P. Ning, Eds., vol. 5789. Springer Berlin Heidelberg, 2009, pp. 355–370.

[14] J. Xu and E.-C. Chang, "Towards efficient proofs of retrievability," in Proceedings of the 7th ACM Symposium on Information, Computer andCommunications Security, ser. ASIACCS '12. New York, NY, USA: ACM, 2012, pp. 79–80.

[15] E. Stefanov, M. van Dijk, A. Juels, and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks," in Proceedings of the28th Annual Computer Security Applications Conference, ser. ACSAC '12. New York, NY, USA: ACM, 2012, pp. 229–238

[16] 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. Kutyłowskiand J. Vaidya, Eds., vol. 8712. Springer International Publishing, 2014, pp. 239–256.

[17] J. Li, X. Tan, X. Chen, and D. Wong, "An efficient proof of retrievabilitywith public auditing in cloud computing," in 5th International Conferenceon Intelligent Networking and Collaborative Systems (INCoS), 2013, pp. 93–98.

[18] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplicationwith efficient and reliable convergent key management," IEEETransactions on Parallel and Distributed Systems, vol. 25, no. 6, pp. 1615–1625, June 2014.

[19] R. Di Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," in Proceedings of the 7th ACMSymposium on Information, Computer and Communications Security, ser. ASIACCS '12. New York, NY, USA: ACM, 2012, pp. 81–82.

[20] W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage," in Proceedings of the 27th Annual ACM Symposiumon Applied Computing, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 441–446.

[21] J. Douceur, A. Adya, W. Bolosky, P. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in 22nd International Conference on Distributed Computing Systems, 2002, pp. 617–624.

[22] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryptionand secure deduplication," in Advances in Cryptology – EUROCRYPT2013, ser. Lecture Notes in Computer Science, T. Johansson and P. Nguyen, Eds. Springer Berlin Heidelberg, 2013, vol. 7881, pp. 296–312..

