International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 01
January 2015

Secure Auditing and Deduplicating Data in Cloud Computing

M Madhuri latha[1], Shabaaz Shaik[2] , K Rama Krishniah[3],D Sarath Babu[4]
[1]M.Tech Student, Dept of CSE, R K College of Engineering, Krishna-521456,A. P., India
[2]Assistant Professor, Dept of CSE, R K College of Engineering, Krishna-521456,A. P., India
[3]Professor & Principal, Dept of CSE,R K College of Engineering, Krishna-521456,A. P., India
[4]Assistant Professor, Dept of CSE, N V R College of Engineering, Krishna-522201,A. P., India

ABSTRACT: As the cloud computing technology develops during the last few years, 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 understand 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.

Index Terms—Deduplication, authorized duplicate check, Auditor, confidentiality, hybrid cloud.

INTRODUCTION:

Cloud storage is a representation 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 cut down 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 accomplish 40 trillion gigabytes in 2020. Even though cloud storage system has been widely adopted, it fails to accommodate some important emerging needs such 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 (or block) and make a link to the file (or block) for every client who owns or asks to store the same file (or block). Unfortunately, this action of deduplication would lead to a number of threats potentially affecting the storage system , for

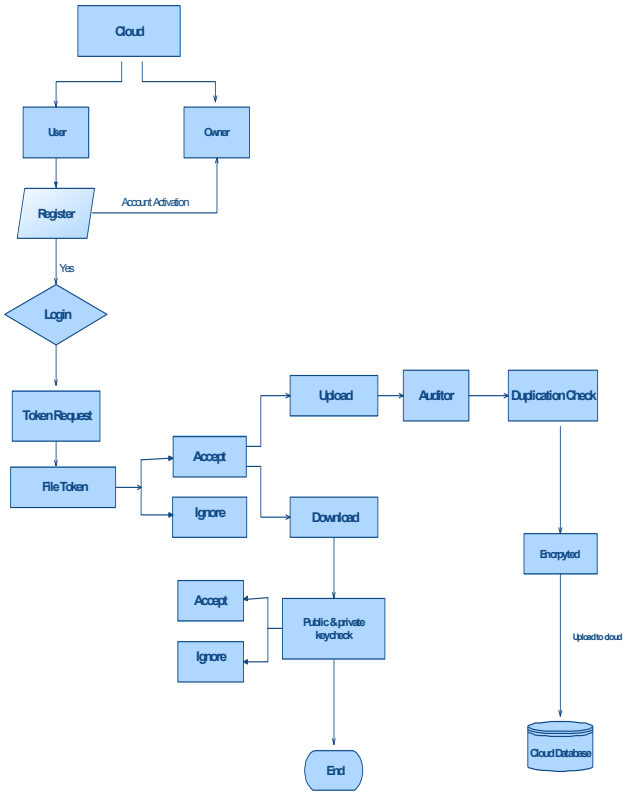
International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 01
January 2015

example, a server telling a client that it (i.e., the client) 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 (or block of data) is solely based on a static, short value (in most cases the hash of the file) . Thus, the second problem is generalized as how can the cloud servers efficiently confirm that the client (with a certain degree assurance) owns the uploaded file (or block) before creating a link to this file (or block) 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 a 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 SecCoud 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. In order to prevent the leakage of such side channel information, we follow the tradition of and design a proof of ownership protocol between clients and cloud servers, which allows clients to prove to cloud servers that they exactly own the target data.

SYSTEM ARCHITECTURE

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



SYSTEM ANALYSIS

EXISTING SYSTEM:

Ateniese et al. proposed a dynamic PDP schema but without insertion operation.Erway et al. improved

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 01
January 2015

Ateniese et al.'s work and supported insertion by introducing authenticated flip table.Wang et al.proposed proxy PDP in public clouds.Zhu et al. proposed the cooperative PDP in multi-cloud storage.Wang et al. improved the POR model by manipulating the classic Merkle hash tree construction for block tag authentication.Xu and Chang proposed to improve the POR schema with polynomial commitment for reducing communication cost. Stefanov et al. proposed a POR protocol over authenticated file system subject to frequent changes. Azraoui et al. 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. considered a new cloud storage architecture with two independent cloud servers for integrity auditing to reduce the computation load at client side.

DISADVANTAGES OF EXISTING SYSTEM: 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.

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. 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 (i.e., the client) 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 (or block of data) is solely based on a static, short value (in most cases the hash of the file).

PROPOSED SYSTEM: 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 maintenance of a MapReduce cloud, which aids clients produce data tags before uploading as well as audit the integrity of data having been stored in cloud.Besides supporting integrity auditing and secure deduplication, SecCloud+ enables the guarantee of file confidentiality.We propose a method of directly auditing integrity on encrypted data.

ADVANTAGES OF PROPOSED SYSTEM:
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 SecCoud is supported on both block level and sector level. In addition, SecCoud also enables secure deduplication.The challenge of deduplication on encrypted is the prevention of dictionary attack. Our proposed SecCloud system has achieved both integrity auditing and files deduplication.

RELATED WORK

Since our work is related to both integrity auditing and secure deduplication, we evaluate 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. 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 , several works concerned on how to realize PDP on dynamic scenario: Ateniese et al. proposed a dynamic PDP schema but without insertion operation; Erway et al. improved Ateniese et al.'s work and supported insertion by introducing authenticated flip table; A similar work has also been contributed in . Nevertheless, these proposals suffer from the computational overhead for tag generation at the client. To fix this issue, Wang et al. proposed proxy PDP in public clouds. Zhu et al. proposed the cooperative PDP in multi-cloud storage. Another line of work supporting integrity auditing is proof of retrievability (POR) . Compared with PDP, POR not merely assures the cloud servers possess the target files, but also guarantees their full recovery. In ,clients apply erasure codes and generate authenticators for each block for verifiability and retrievability. In order to achieve efficient data

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 01
January 2015

dynamics, Wang et al. improved the POR model by manipulating the classic Merkle hash tree construction for block tag authentication. Xu and Chang proposed to improve the POR schema in with polynomial commitment for reducing communication cost. Stefanov et al. proposed a POR protocol over authenticated file system subject to frequent changes.

B. Secure Deduplication

Deduplication is a technique where the server stores only a single copy of each file, regardless of how many clientsasked 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. introduced the proof of ownership protocolwhich lets a client efficiently prove to a server that thatthe client exactly holds this file. Several proof of ownershipprotocols based on the Merkle hash tree are proposed toenable secure client-side deduplication. Pietro and Sorniotti proposed an efficient proof of ownership scheme bychoosing the projection of a file onto some randomly selectedbit-positions as the file proof.Another line of work for secure deduplication focuses onthe confidentiality of deduplicated data and considers to makededuplication on encrypted data. Ng et al. firstly introducedthe private data deduplication as a complement of publicdata deduplication protocols of Halevi et al. Convergentencryption is a promising cryptographic primitive forensuring data privacy in deduplication. Bellare et al. formalized this primitive as message-locked encryption, andexplored its application in space-efficient secure outsourcedstorage. Abadi et al. further strengthened Bellare et al'ssecurity definitions by considering plaintext distributionsthat may depend on the public parameters of the schemas.Regarding the practical implementation of convergent encryptionfor securing deduplication, Keelveedhi et al. designedthe DupLESS system in which clients encrypt under file-basedkeys derived from a key server via an oblivious pseudorandomfunction protocol.

C. **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.

Implementation

Cloud Service Provider: In this module, we develop Cloud Service Provider module. This is an entity that provides a data storage service in public cloud. The CS provides the data outsourcing service and stores data on behalf of the users. To reduce the storage cost, the CS eliminates the storage of redundant data via deduplication and keeps only unique data. In this paper, we assume that CS is always online and has abundant storage capacity and computation power.

Data Users Module: A user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users. In the authorized deduplication system, each user is issued a set of privileges in the setup of the system. Each file is protected with the convergent encryption key and privilege keys to realize the authorized deduplication with differential privileges.

Auditor : 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 first design goal of this work is to provide the capability of verifying correctness of the remotely stored data. public verification, which allows anyone, not just the clients originally stored the file, to perform verification.

Secure De-duplication System: We consider several types of privacy we need protect, that is, i) unforgeability of duplicate-check token: There are two types of adversaries, that is, external adversary and internal adversary. As shown below, the external adversary can be viewed as an internal adversary without any privilege.

If a user has privilege p, it requires that the adversary cannot forge and output a valid duplicate token with any other privilege p' on any file F, where p does not

match p'. Furthermore, it also requires that if the adversary does not make a request of token with its own privilege from private cloud server, it cannot forge and output a valid duplicate token with p on any F that has been queried.

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple.

OBJECTIVES: Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user
will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN: A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements..Select methods for presenting information..Create document, report, or other formats that contain information produced by the system.The output form of an information system should accomplish one or more of the following objectives.Convey information about past activities, current status or projections of the Future. Signal important events, opportunities, problems, or warnings. Trigger an action. Confirm an action.

CONCLUSION: Aiming at achieving both data integrity and deduplication in cloud, we propose SecCloud 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. In addition, SecCoud enables secure deduplication through introducing a Proof of Ownership protocol and preventing the leakage of side channel information in data deduplication. 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 deduplication directly on encrypted data.

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: Serveraided 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]. 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 *Proceedings 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 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 Transactions on 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 and Application 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 and Communications 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 the 28th 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. Kutyłowski and 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 retrievability with public auditing in cloud computing," in *5th International Conference on 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 deduplication with efficient and reliable convergent key management," *IEEE Transactions 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 ACM Symposium 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 Symposium on 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 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.

[23] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," in *Advances in Cryptology – CRYPTO 2013*, ser. Lecture Notes in Computer Science, R. Canetti and J. Garay, Eds. Springer Berlin Heidelberg, 2013, vol. 8042, pp. 374–391.

[24] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology — CRYPTO 2001*, ser. Lecture Notes in Computer Science, J. Kilian, Ed. Springer Berlin Heidelberg, 2001, vol. 2139, pp. 213–229.