Protective Auditing and Deduplicating Data in Cloud

Munde Sadashiv C.
PG Student,
MBES College of Engineering,
Ambajogai

B. M. Patil
PG Dept,
MBES College of Engineering,
Ambajogai

ABSTRACT

As the cloud computing becomes more popular for the virtually storing data because it provides the big storage space as well as provides the portability and mobility functions. Whenever we are moving around the world we can access the cloud data by setting up internet connection in between our device and cloud server. Cloud decreases the burden of storing and handling the huge data. External cloud storage raises the issues of reliability and security concerns. Some of the issues about deduplication and integrity auditing are in existing system. Major objective of this work is to acquire the data integrity and deduplication in order to keep our system secure as well as healthy. In previous work there is reduced the computation time of uploading and downloading time of data with the help of map reduce technique. As the customers requirement data is being more secure so here it use the encryption strategy for more security.

Keywords

Cloud Client, Cloud Server, and Auditor.

1. INTRODUCTION

Now a day's cloud becomes an attractive trend toward the people. Cloud provides the space for user to store data in virtualized pool storage which may be accessible wherever you want with the help of internet capable device. This portability as well as the scalable services provided by the cloud affect that peoples preferred to store their personal data on cloud storage. Cloud also beneficial for cost saving .It provides the resource sharing feature. There are some emerging needs which fail by cloud such as auditing integrity of files by client and detecting the duplicate files of data by cloud server. Cloud server minimizes the heavy load of storing and maintaining data stored on cloud. As compares to traditional storage cloud storage store the data at uncertain storage domain without any control of client. This grows the security concerns of the client data. Cloud storage is susceptible to security threats from both outside and inside of the cloud [1].Some clouds hides the loss of data from client in order to maintain their reputation. Another one more serious problem is that in order to save the space as well as money some cloud discards the rarely used data of ordinary clients. Here the issue arises how to recover these problems of integrity verification of client data. Second problem is that secure deduplication. Most of the peoples uses the cloud and upload the data on cloud and this activity gives results of duplicates of the same data/files on server which exceeds the storage of cloud. Then here needs the cloud to store only deduplicate data that is keep only the single copy of data on cloud. If any user request to access same file which already on server giving him/her the reference of that file and avoid duplication of same file. This action of deduplicating files may lead to security threats [3][2].Some potential security threats are arises while client upload the file towards the cloud that cloud tell that file is already exist on cloud these files are sensitive sometimes . From these types of attack client should solely get that file which is permitted to him/her [3].Second problem is that cloud server efficiently confirm that client owns the uploaded file by creating link to that file. Here aim is that to auditing integrity and dedulication .This can be implement by using SecCloud and SecCloud. SecCloud used for secure cloud. It fixes the issue of previous work of heavy load of computational time. SecCloud generates the tags which we upload to the cloud. It also prevents the leakage of information side channel. SecCloud+ it uses the encryption so data becomes more confidential on cloud. SecCloud uses convergent key encryption in order to defend to the attacker. It prevents dictionary attacks [4].

1.1 Objectives

Integrity Auditing: It verifies the accuracy or the correctness of data uses verification method that is public verification and stateless verification. In public verification it allows anyone not only the client for verification. Stateless verification eliminates the state information maintenance. Secure Deduplication: The process which used for keep the single copy of file in order to save space on cloud duplicate file not allowed to store on cloud by performing duplicate check. Cost Effective: It is does not take any extra cost to download as well as uploading file operation. Also it does not change the upload download mechanism.

2. LITERATURE REVIEW

The main goal of this this work is to acquire data integrity as well as the secure data deduplication. In existing system there was only the single server was used but in proposed system Cloud uses the multiple servers in order to load balance as well better uploading downloading speed. Also Data reliability is actually a very critical issue in a deduplication storage system because there is only one copy for each file stored in the server shared by all the owners.

2008: Shacham et al. [5] proposed scheme, built from BLS signatures and secure in the random oracle model, features a proof-of-retrievability protocol in which the client's query and server's response are both extremely short. This scheme allows public verifiability: anyone can act as a verifier, not just the file owner. 2009: Wang et al.[13] study the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud.

2010: Armbrust et al. [1] Cloud storage has been increasingly prevalent because of its advantages [1]. Currently, commercial cloud storage services including Microsoft Sky drive, Amazon S3 and Google Cloud Storage have attracted millions of users.2011: Halevi et al. [3] proposed to identify attacks that exploit client-side deduplication, allowing an attacker to gain access to arbitrary-size files of other users based on very small hash signatures 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.

2011: Ateniese et al. [6] introduced a model for provable data possession (PDP) that can be used for remote data checking: A client that has stored data at an untrusted server can verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs.

2012: Zhu et al. [11] addressed the construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration, in which work consider the existence of multiple cloud service providers to cooperatively store and maintain the clients' data.2012: Ng et al. [20] proposed a new notion which is called private data deduplication protocol, a deduplication technique for private data storage is introduced and formalized.2013: Li et al. [17] study the problem of ensuring the integrity of data storage in Cloud Computing. To reduce the computational cost at user side during the integrity verification of their data, the notion of public verifiability has been proposed.2013: 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.

2013: Ristenpart et al. [22] provide definitions both for privacy and for a form of integrity that is called tag consistency. Based on this foundation, we make both practical and theoretical contributions. 2013: Yuan et al. [2] solved problem with a novel scheme based on techniques including polynomial-based authentication tags and homomorphic linear authenticators. 2013: Bellare et al. [4] formalized this primitive as message-locked encryption, and explored its application in space efficient secure outsourced storage.2014: Chen et al. [18] attempt to formally address the problem of achieving efficient and reliable key management in secure deduplication. We first introduce a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud.

3. EXISTING SYSTEM

- There are many strategies for deduplication is proposed. Server side deduplication, client side deduplication and file level, block level. These are the stratagies where deduplication can be done.
- Bellare et al introduced message locked encryption as well as its application.
- Li shown the KeyManagement issue in block-level deduplication by sending these keys over multiple servers after encrypting the files. Bellare et al showed how to protect data confidentiality.
- The initial problem is integrity auditing. The cloud server minimizes the huge load from users to storing and maintaining those bulk amounts of data. The difference is that in traditional storage and cloud storage as shown below. Cloud refers to use of internet to transfers the file from client to uncertain space which is not under the control of user. This result in increasing the client integrity of their data.
- The second problem is secure deduplication. In the recent days many users prefer cloud storage to store their data on cloud. This result in large amount of duplicate data on cloud. This becomes the issue of storage space as well as to maintain it. By the EMC survey there is 75% of the digital data on cloud are duplicated file.
- Actually, this action of deduplication would cause to many threats potentially affecting the storage system, for example if there is already exist file and client uploading same file that time server suggest to client the file is already existed do not need to upload the same file. This file may be sensitive some times.

3.1 Problem statement

In problem statement as the existing system having problem with auditing integrity because of huge data on cloud server and user requirement of accurate data retrieve from cloud not whole data. Here it solved this problem in proposed system.
Also another one problem of deduplication of data was raised due to increasing growth of duplicate files on cloud these duplicate files leads to exceed the storage space as well as more time consume for computing.

4. PROPOSED SYSTEM

In this paper, this work designs the system that is capable to achieve more reliability as compare to traditional cloud. With the help of distributed server it can achieve data confidentiality here; It split the file into number of parts these parts are stored on different servers instead of single server.

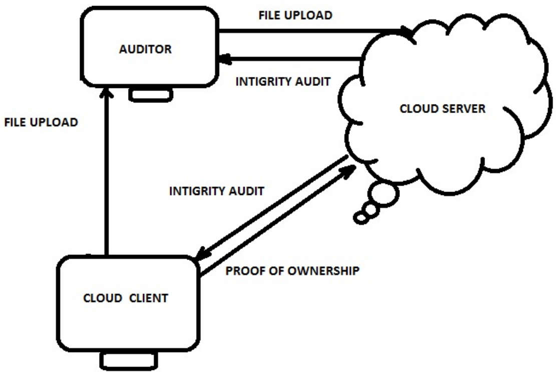

Fig 1: Secure Cloud Architecture

This work determines that proposed SecCloud system has achieved both integrity auditing and files deduplication. However, it is not possible to hide contents from the server in simple words there this work imposed the functionality of integrity auditing and deduplication on the plain paper. This work performs the integrity auditing on encrypted file. So as the name given to the system is seccloud system, in this section of seccloud it provides the key server and distributes the key to its client. Here we are just adding extra features to the previous work that is encryption strategy added. Seccloud involves the protocols as file upload protocol, integrity audit protocol and proof of ownership protocol. File uploading protocol uses more steps to take communication in between cloud client and the key server. The client here required the convergent key from key server which require for encrypting the file at the uploading time.

4.1 Convergent key Encryption

Convergent key encryption provides the data confidentiality in deduplication.Using convergent key data will be encrypted and to detect the duplicate data there we use tag generation tags are used when the data are same then tags would be same.

- 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 cipher text ctF;
- Decrypt(ckF; ctF) : The decryption algorithm receive the

  Convergent key ckF and cipher text ctF that is input files cipher text as well as output file cipher text.
- TagGen(F): It takes the content of the file as input and generates the tags of those contents identified as TagF.

4.2 Advantages

- In this work here achieved the data integrity as well as data deduplication and data inconsistency.
- Due to deduplication space availability can be increased.
- It achieves reliability, data confidentiality and releases the heavy burden from user.

5. RESULT AND DISCUSSION

In existing system there is not used encryption key management server. Proposed system defines that encryption and implements this work by using encryption at the client level. This encryption gives user more confidentiality for file storage on the cloud. As shown below the table contains the uploading and downloading parameters of file. It contains the file size, time for uploading file and time for downloading file. As if file contains duplicate data it will take more time to upload file as well as to download file rather than deduplicate file. Analysis from this work is that it can decrease the file upload and download time of the file as well it save the storage space. It provides high reliability of files due to file encryption mechanism.

Table 1: Analysis of file uploading and downloading time of duplicate files

File Size(in KB)	Time for Uploading file (in ms)	Time for Downloading File(in ms)
1	1.5	1
2	2	1.5
3	2.5	2
4	3	2.5

According to given table parameters of file uploading as well as downloading time we can easily calculate through the analysis of above parameters values. As the file containing duplicate data so time taken for upload as well as download is more as compare to deduplicate files.

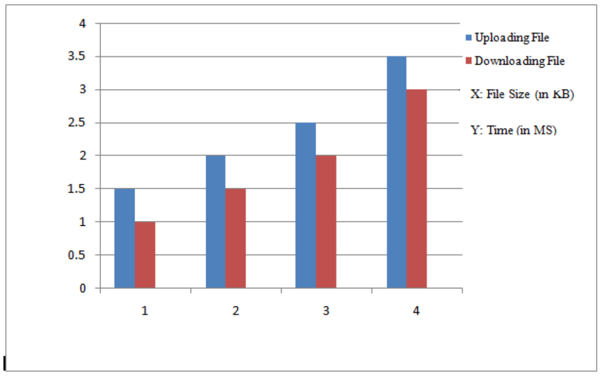

Fig 2: Time and space complexity without deduplication.

As shown in Fig.2 the uploading and downloading time of files given as the file sizes are in kb(kilo bytes) on x axis and the time for uploading file is given on y axis in ms (mili seconds) without deduplication file uploading time increases.

Table 2: Analysis of file uploading and downloading time of deduplicate files

File Size(in KB)	Time for Uploading file (in ms)	Time for Downloading File(in ms)
1	1	0.5
2	1.7	1.3
3	2	1.7
4	3	2.3

As shown in table 2 there are different parameters given for analysis of time for uploading and downloading the file using deduplication. As compare with duplicate files time required to upload as well as download is less in time. This is most important feature of data deduplication.

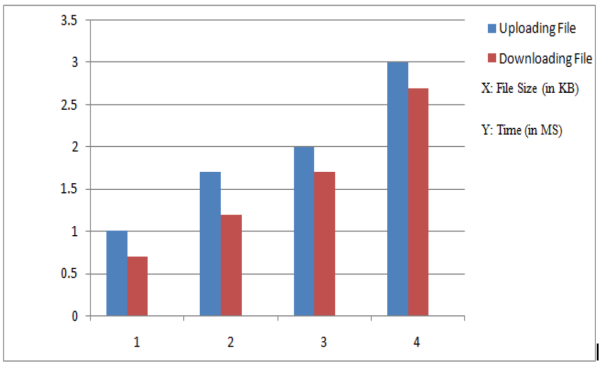

Fig 3: Time and space complexity with deduplication.

As shown in Fig.3 file uploading time by using deduplication is less as compare with duplicate file.

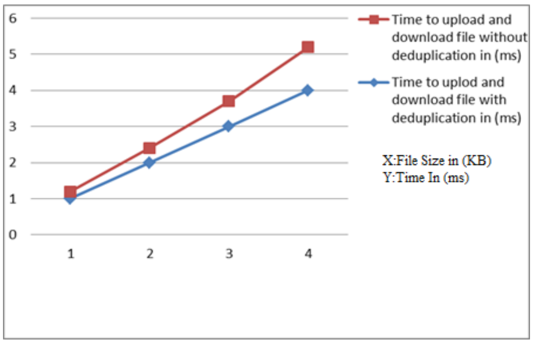

Fig 4: Comparison graph of duplicate and deduplicate files for uploading and downloading time.

As shown above Fig.4 there is comparison graph of file uploading time with duplication and without deduplication. Comparison of uploading time and downloading time is easily analyzed with the help of comparison graph.

6. CONCLUSION AND FUTURE WORK

To achieve the data integrity as well as data deduplication here it introduces two models that are given .First one is SecCloud and seccloud+. SecCloud is useful for client which generates the tags of source file. It introduces the proof of ownership protocol for avoiding the leakage of side channel information. As compare to previous work computation time is decreased here. SecCloud+ uses the encryption mechanism for stored file. In simple words it stores all files in encrypted format.

7. ACKNOWLEDGEMENT

I would like to sincerely express thanks to my guide Dr. B. M. Patil for his appreciable support, continuous encouragement and his invaluable suggestions. I am grateful for all the suggestions and hints provided by him. It was great moment to work with him. As part of future work, SecCloud+ should be extended with more security features such as proofs of retrievability, data integrity checking and search over encrypted data.

8. REFERENCES

[1] 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.

[2] 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.

[3] 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.

[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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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

[10] 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.

[11] 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.

[12] T. Ristenpart, studied the detailed of “Message-locked encryption and secure deduplication,” Eds. Springer Berlin Heidelberg, 2013

[13] 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.