ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

ROBUST EFFICIENT STORAGE MANAGEMENT USING DEDUPLICATION DATA IN CLOUD

PUSHPALATHA.A #1, MAHALAKSHMI.C#2
#1 ME-Second year , Department of Computer Science and Engineering,
Bharathiyar Institute of Engineering for Women,
Deviyakurichi(po),Attur(tk),Salem(dt),Tamilnadu, India.
469lathacsebiew@gmail.com
#2 Assistant Professor, Department of Computer Science and Engineering,
Bharathiyar Institute of Engineering for Women,
Deviyakurichi(po),Attur(tk),Salem(dt),Tamilnadu, India.
maha.it2008@gmail.com

ABSTRACT

As the cloud deducting knowledge advances through the last time, subcontracting data to cloud package for storing becomes an nice-looking trend, which profits in sparing efforts on heavy data conservation and controlling. However, since the subcontracted cloud package is not fully truthful, it raises safety concerns on how to understand data deduplication in cloud while achieving reliability checking. In this work, we study the problem of reliability checking and protected deduplication on cloud data. leveling at succeeding both data checking and deduplication in cloud, we propose two protected systems, namely SecCloud and SecCloud+.SecCloud introduces an checking entity with a maintenance of a MapReduce cloud, which helps clients generate data codes before uploading as well as audit the integrity of data having been stored in cloud. Related with earlier work, the computation by user in SecCloud stands critically reduced through the file uploading and checking points. SecCloud+ is considered encouraged by the information that consumers always want to encode their data before uploading, and enables dependability checking and protected deduplication on encoded data.

1.INTRODUCTION

Cloud package is a classical of interacted creativity package where data is stored in virtualized groups of package which are commonly presented by third parties. Cloud package provides clients with profits, reaching from cost save and beginner's suitability, to mobility chances and save.

These unlimited features interest more and more consumers to use and package their private data to the cloud package: allowing to the study report, the size of data in cloud is estimated to succeed 40 trillion gigabytes in 2020. Even though cloud package system has been generally assumed, it fails toward provide accommodations some essential initial needs such as the facilities of testing integrity of cloud files by cloud users and distinguishing duplicated files by cloud servers.

The first problematic is integrity checking. The cloud server is able to relieve clients from the heavy problem of package controlling and conservation. The most variance of cloud package from out-of-date in-house package is that the data is transmitted via Internet and stored in an unreliable area, not under control of the clients at all, which unavoidably increases clients great disquiets on the integrity of their data. These disquiets invent from the fact that the cloud package is vulnerable to confidence dangers from both outside and inside of the cloud, and the unrestrained cloud servers may unreceptively hide some data loss instances from the clients to conserve their status. What is more thoughtful is that for saving money and space, the cloud servers might even dynamically and intentionally remove irregularly accessed data files fitting to an usual

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

client. Considering the large size of the subcontracted data files and the clients' constrained source skills, the first problem is generalized as how can the client powerfully perform quarterly integrity authentications even without the local copy of data files.Second problem is secure deduplication. The rapid acceptance of cloud services is attended by increasing sizes of data stored at inaccessible cloud servers. Mid these faraway stored files, most of them are imitative: allowing to a recent survey by EMC, 75% of modern numerical data is duplicated copies. This fact raises a knowledge specifically deduplication, in which the cloud servers like to deduplicated by trust only a single copy for each file and make a link to the file (or block) for every client who owns or asks to store the same file .Inappropriately, this stroke of deduplication would lead to a number of pressures possibly touching the package 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 complex occasionally. These rounds start from the goal that the evidence that the client owns a given file is only based on a static, small cost Thus, the second problem is generalized as how can the cloud servers powerfully check that the client owns the uploaded file before creating a bond to this file.

2.EXISTING SYSTEM

SecCloud introduces an checking object with a management of a Map Reduce cloud, which profits users create data codes already uploading as well as check the integrity of data having been stored in cloud. This design shots the issue of earlier work that the computational load at user or auditor is too massive for code generation. For totality of fine-grained, the functionality of checking planned in SecCloud is maintained on both block level and sector level. In addition, SecCloud also supports secure deduplication. Communication that the "security" measured in SecCloud is the avoidance of drip of side channel evidence. In order towards stop the drip of such side channel material, we follow the convention of and design a proof of ownership protocol between clients and cloud servers, which permits clients to prove to cloud servers that they accurately private the aim data.

Encouraged by the fact that clients continuously want to encode their data before uploading, for details ranging from personal privacy to trade policy, we present a key server into SecCloud as with and suggestion the SecCloud+ schema. Further assistant reliability checking and secure deduplication, SecCloud+ allows the promise of file secrecy. Exactly, thanks to the goods of deterministic encryption in convergent encryption, we suggest a method of nonstop checking integrity on encoded data. The task of deduplication on encoded is the avoidance of thesaurus dose. As with, we make a variation on convergent encryption such that the convergent key of file is generated and exact by a secret "seed", such that any challenger could not straight arise the convergent key from the comfortable of file and the dictionary attack is prevented.

3.PROPOSED SYSTEM

SecCloud system has achieved both reliability checking and file deduplication. However, it cannot prevent the cloud servers from meaningful the comfortable of files consuming been deposited. In other words, the functionalities of reliability checkingand protected deduplication are only imposed on plain files. In this section, we propose SecCloud+, which allows for reliability checking and deduplication on encrypted files.

Cloud Clients have large data files to be stored and rely on the cloud for data conservation and subtraction. They can be either individual consumers or commercial organizations;

Cloud server virtualize the properties allowing to the requests of clients and expose them as package pools. Typically, the cloud clients may buy or lease package capacity from cloud servers, and store their separable data in these believed or chartered places for prospect operation. Auditor which helps clients upload and audit their outsourced data maintains a MapReduce cloud and acts like a license authority. This assumption presumes that the auditor is connected with a brace of municipal and

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

remote keys. Its public key is made available to the other entities in the system

The design aim of file privacy need towards avoid the cloud servers from accessing the content of files. Specially, we require that the goal of file privacy needs to be resistant to "dictionary attack". That is, even the opponents have pre-knowledge of the "dictionary" which includes all the possible files, they still cannot recover the target file.

4.METHODOLOGY DESCRIPTION

4.1 File Confidentiality

The proposal goal of file privacy needs to prevent the cloud servers from retrieving the comfortable of files. Particularly, we require that the goal of file privacy needs to be strong to "dictionary attack". That is, even the opponents have pre-knowledge of the "dictionary" the promising files, they still cannot recover the target file

4.2 Protected Deduplication

Deduplication is a technique where the server provisions only a one copy of each file, irrespective of how many clients asked to stock that file, such that the disk space of cloud servers as well as web bandwidth are saved. Nevertheless, small client side deduplication leads to the leakage of side channel information. For example, a server powerful a consumer 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.

4.3 Encryption & Decryption

Encryption and decryption provides data confidentiality in deduplication. A user derives a convergent key from the data content and encrypts the data copy with the convergent key. The user springs a tag for the data copy, such that the tag will be used to detect duplicates. Here, we accept that the code truth belongings holds, i.e., if two data copies are the same, then their tags are the same., a convergent encryption structure can be defined with four primitive functions:

encryption scheme can be defined with four primitive functions:

4.4 Integrity Checking

The first design goal of this work is to provide the ability of validating accuracy of the slightly warehoused data. The integrity authorization further requires two features:

1. Public verification, which permits someone, not just the clients originally stored the file, to perform verification;
2. Stateless verification, which is capable to eliminate the need for state information maintenance at the verifier side between the actions of checking and data package.

5. ARCHITECTURE DIAGRAM

Fig .1.SecureCloud Architecture

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

• Cloud Clients have A large data files to be stored and
rely on the cloud for data conservation and subtraction.
They can be whichever single customers or profitable
organizations;
• Cloud Servers virtualize the resources according to the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

requirements of clients and expose them as package
pools. Typically, the cloud clients may buy or lease
package capacity from cloud servers, and store their
individual data in these bought or rented spaces for
• Checker which helps users upload and audit their outsourced data maintains a MapReduce cloud and actions like a license specialist. This statement believes that the
auditor is associated with a pair of public and remote keys. Its public key is complete accessible to the further entities in the system.
The SecCloud system supporting file-level deduplication
includes the following three protocols respectively highlighted by red, blue and green
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 package 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 package 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) Reliability checkingProtocol*:* 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.
However, trivial client side deduplication leads to the leakage of side channel information. For example, a server powerful a user that it need not send the file reveals that some other client has the particular same file, which could be complex evidence in some case.derives a convergent key from the data content and codes the data copy with the convergent key. In addition, the user derives a tag for the data copy, such that the tag.

6.CONCLUSION

Aiming at completing both data integrity and deduplication in cloud, we propose SecCloud and SecCloud+. SecCloud introduces an checking entity with maintenance of a MapReduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of files consuming been deposited in cloud. In addition, SecCloud enables protected deduplication over presenting a Proof of Proprietorship protocol and avoiding the leakage of side channel information in data deduplication. Compared with earlier work, the subtraction by user in SecCloud is greatly reduced during the file uploading and checking phases. SecCloud+ is an advanced construction 1motivated by the fact that customers always want to encrypt their data before uploading, and allows for reliability checkingand protected deduplication directly on encrypted data.

7.FUTURE ENHANCEMENT

SecCloud enables protected deduplication through introducing a Proof of Ownership protocol and preventing the leakage of side network evidence in files deduplication. The computation by user in SecCloud is greatly reduced during the file uploading and checking phases. SecCloud+ is an advanced construction motivated by the fact that customers always want to encrypt their data before uploading, and allows for reliability checking and protected deduplication directly on encrypted data.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

[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 ofownership 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.

[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 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, andJ.-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,

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
TET)
Vol. 3, Special Issue 2, March 2016

"Compact proofs of retrievability," inProceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology,ser. 2008, pp.90–107.