IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-3, Issue-10 E-ISSN: 2347-2693

Secure Authorized Deduplication Based on Hybrid Cloud Approach

J. Suresh[1*] and S. Jhansi Rani[2]

[1*,2]*Department of Computer Science and Systems Engineering, Andhra University, India*

www.ijcseonline.org

Received: Sep/22/2015 Revised: Oct/26/2015 Accepted: Oct /26/2015 Published: Oct /31/ 2015

***Abstract*—** Data de-duplication is a data compression technique which is used to eliminate the duplicates for the repeated data. It is been extensively used in the cloud storage to minimize the amount of storage size and save maximum bandwidth. The convergent technique is been suggested, to encrypt the data before outsourcing to protect the confidentiality for the sensitive data while using de-duplication. In order to provide better data security, a hybrid cloud is been proposed. A hybrid cloud is a combination of private cloud and public cloud bound together. Private cloud is a place where some critical activities can be performed like security, setting the privileges to the users, setting the token number etc. whereas the public cloud is a place where some noncritical activities are performed like outsourcing an encrypted file. This paper formally addresses authorization with data de-duplication; we present several new de-duplications supporting the authorized duplicate check in the hybrid cloud architecture. In this paper we attempt to address the authorized de-duplication check, combined with convergent encryption for providing the better security to the sensitive data using the hybrid cloud architecture.

***Keywords*—** Deduplication; authorized duplicate check; confidentiality; hybrid cloud

I. INTRODUCTION

Cloud computing has been widely spread in the world, it is an emerging style of computing where the data, applications and resources are provided to the users as services over the web. The services provided may be available globally, low on cost, on demand, scalable, pay-as-you-grow. One of the critical challenges for cloud storage services is management of the duplication. It has attracted more and more attention at recent times. In data storage to reduce the data copies we can go for duplication techniques [1]. The data compression technique is been used for eliminating the duplicate copies of the repeated data in the cloud storage. This technique is used to the network data transfers to minimize the number of bytes that is been sent and improve the storage utilization. Deduplication eliminates the unnecessary data by maintaining only one physical copy and pointing the other repeated data to that copy [5]. Data de-duplication occurs in file level as well as block level. Same file duplicate copies will be eliminated in the file level de-duplication and in the non-identical files, the data blocks that occurred will be eliminated with the block level de-duplication.

Although data de-duplication brings lot of advantages, security and privacy concerns may arise as the users sensitive data may be susceptible to the insider attacks as well as an outsider attack [18]. The traditional encryption requires different users to encrypt data with their own keys. Then identical copies from multiple users will lead to different cipher texts, making de-duplication more impossible[9]. One of the new technique is been proposed for data confidentiality and de-duplication feasible is to use convergent encryption [2]. This convergent encrypt provides one convergent key to encrypt/decrypt the data, which is obtained from the cryptographic hash value of the content from the data copy. After completion of key generation and data encryption, users send the cipher text data to the cloud. The proof of ownership protocol is used to issue the proof to the storage server, that the user indeed owns a particular file. This is done to prevent unauthorized access [7]. A pointer from the server will provide to user, after the proof submission, the user who is having the subsequent file without needing to upload the same file again [4]. The encrypted file can be downloaded by the user and also decrypted by the corresponding data users with their convergent keys. In this way convergent encryption permits the cloud to perform de-duplication on the cipher texts and use proof of ownership to restrict the unauthorized user from accessing the file [3].

The previous de-duplication system does not support the differential authorization and the duplicate check, which is an important security aspect in many of the applications. In the authorized de-duplication system each user is assigned with set of privileges during the system start up [8]. Each file uploaded to the cloud is associated with a set of privileges assigned, which specifies what kind of user is allowed to use the duplicate check and access the files. Before accepting a duplicate check request for a file, the user needs to take his/her file and his/her own privileges as inputs [6]. Then the user can actually find the duplicate for this file, if only there is a copy of these files that is matched with the privilege stored in the cloud.

Corresponding Author: *J. Suresh, suresh.list@gmail.com*
Department of Computer Science, Andhra University, Visakhapatnam, India

II. PRELIMINARIES

In this section we go through the notations used in this paper and analyzing the secure primitives used in our secure duplication.

A. *Symmetric Encryption*

The Symmetric encryption uses a common secret key for encrypting and decrypting the information.
A symmetric encryption scheme consists of three primitive functions:

- KeyGenSE(1^{λ}) → κ; It is the key generation algorithm that generates a secret key κ using security parameter 1^{λ}.
- EncSE(κ, M) → C; It is the symmetric encryption algorithm that uses the secret key κ that is been initiated and message M, and then outputs the cipher text C; and
- DecSE(κ, C) → M; It is the symmetric decryption algorithm that uses the secret key κ that is been initiated and cipher text C, and then outputs the original message M.

B. *Convergent Encryption*

The convergent encryption provides secure data confidentiality in de-duplication. The data owner gets the convergent key from each of the original data copy and encrypts that data copy with the convergent key. The user also derives a tag from the data copy to detect the duplicates. If any two data copies are identical, then their tags will be same. To identify and detect the duplicates, the user must send a tag to the server side in order to test for the identical copies that have been already stored in the server side. The convergent encryption consists of four primitive functions.

- KeyGenCE(M) → κ; It is the key generation algorithm that maps a data copy M to a convergent key κ;
- EncCE(κ, M) → C; It is the symmetric encryption algorithm that takes the convergent key κ and the data copy M as inputs and then outputs the cipher text C;
- DecCE(κ, C) → M; It is the decryption algorithm that takes the cipher text C and the convergent key κ as inputs and then outputs the original data copy M; and
- TagGen(M) → T(M); It is the tag generation algorithm that maps the original data copy M and outputs a tag T(M).

C. *Proof of Ownership*

The proof of ownership (POW) enables the users to prove that they are owner of the particular data copy to the storage server. The POW is implemented as an interactive algorithm runner by a prover and a verifier. Here the prover is a user and the verifier is a storage server. The verifier derives a value ø(M) from the data copy M. To prove the ownership of the data copy M to the verifier, the prover needs to send ø′ to the verifier such that ø′ = ø(M). The security definition for POW will be almost nearly following the threat model from the content distribution network, where as an attacker doesn't know the entire file, but can accomplices who is having the file. The accomplices follow the "bounded retrieval model", such that they can help the attacker to acquire the file, subject to the restriction that they must send lesser bits than the initial min-entropy of the file to the attacker.

III. PROPOSED SYSTEM

A. *Abbreviations and Acronyms*

The abbreviations and acronyms used in this paper are as shown below.

Acronym	Description
S-CSP	Storage cloud service provider
POW	Proof of Ownership
k_f	Convergent encryption key for file F
H	Hash function

Table 1 Notations used in this paper

B. *Hybrid Architecture for Secure Deduplication*

We implement a system that includes public cloud and private cloud and also the hybrid cloud, which is a combination of both private cloud and the public cloud. In this system we provide the data de-duplication which is been used to avoid the duplicate copies of repeated data. User may upload or download the files into the public cloud, private cloud provides the security for that data i.e., only authorized users can upload and download the files into the public cloud, for that user needs to generate the key and stored that key onto the private cloud. During downloading process, user request to the private cloud for key and then only user can access that particular file.

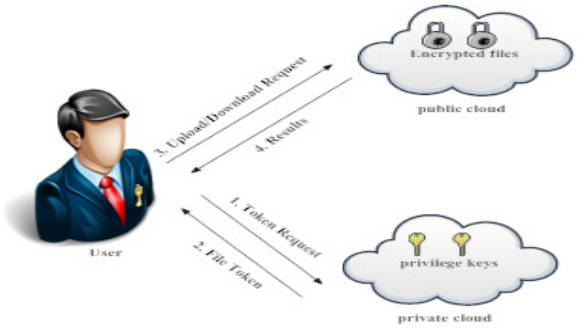

Fig. 1 Hybrid Architecture for Secure Deduplication

C. System Model

The architecture of our system has three modules as shown in fig. 1, those are,

- user
- public cloud
- private cloud.etc.

If the user wants to upload the files on to the public cloud, then the user needs to encrypt that particular file with the convergent key and outsource it to the public cloud, and at the same time the user also generates the key for that particular file and sends that key back to private cloud in order to provide better security. The public cloud uses one algorithm for de-duplication, to avoid the duplicate identical copies of the files which are in public cloud. Hence it can also minimizes the bandwidth i.e., we require the minimum storage space for storing the files on to the public cloud. In the public cloud any unauthorized user can also access/store the data, so we can come to the conclusion, that in the public cloud the data security is not been provided. To provide better security user can make use of the private cloud rather than using the public cloud. The key is generated by the user at the time of uploading a file in the public cloud and stores that key in the private cloud. When user wants to download a file that he/she uploaded, then the request will be sent to the public cloud. Public cloud provides the list of files that are uploaded by many users from the public cloud, where there is no security for the data in public cloud. If user selects one particular file from the list of upload files, then the private cloud sends a message, like enter the key! Then user had to enter the key that is been generated for that particular file. When the user enters the key, private cloud checks whether the key for that particular file is correct or not, then only an access will be given to the particular file to download a file successfully. Hence user can download the file from the public cloud, decrypt that particular file by using the same convergent key which has been used during the encryption of that particular file. In this way user can use this proposed architecture.

IV. ROLES OF ENTITIES

S-CSP: The S-CSP stores the data on behalf of the user. This entity is used as a data storage service in public cloud. The S-CSP eliminates the duplicate identical data copies by using de-duplication technique and keeps the unique data as it is. S-CSP entity is used to minimize the storage cost. S-CSP has ample storage capacity and computational power. When an user sends respective token for accessing his/her file from public cloud, S-CSP matches with the token internally, if it is matched then only he/she sends the file as a cipher text Cf with token, else ways he send abort signal to the user. After receiving the file user can use convergent key KF in order to decrypt the file.

Data User: A user is an entity who wants to access the data copies or files from S-CSP. User generates the key and stores that key into the private cloud. The storage system supporting de-duplication, to upload unique data only but not to upload any duplicate data to save the uploaded bandwidth, which may be maintained by the same user or multiple users. Each file is protected by a convergent encryption key and can access only by authorized user. User needs to register in the private cloud for storing token with corresponding file, which are stored on public cloud. When he/she wants to access that file he/she can access the respective token from private cloud and then can access his/her files from the public cloud. A Token is consisted of file content F and the convergent key KF.

Private Cloud: In order to provide better security, user can make use of the private cloud rather than using the public cloud. User stores the generated key in the private cloud. During the time of downloading system will query for the key to download a file. User can not store the secrete key inside, so for providing security to the key we makes use of the private cloud. Private cloud only stores the convergent key with corresponding file. When user wants to access the key he/she first checks authority of the user then only provides the key.

Public Cloud: Public cloud entity is the place where the users can use for storage purpose. User can upload/download the files in the public cloud. Public cloud is similar to S-CSP. When user wants to download the files from public cloud, it will cross-examine the key which is been generated or stored in the private cloud. When the user's key is matched with the files key at that time only user can download the file, without the key user can not access the file. Authorized user only can access the file. In the public cloud each and every file should be stored in an encrypted format. Hence, we can ensure that there is no chance of unauthorized person hacking our file, but without the convergent key he/she couldn't access the original file. On public cloud there are bunch of files stored, each user access its corresponding file if it's token matches with S-CSP server token

A. Operations performed on Hybrid Cloud

The following operations are been performed on the new de-duplication system.

File Uploading: When the user wants to upload a file to the public cloud, then user should first encrypt the file that is to be uploading, by making use of the symmetric key and move it to the public cloud. At the same time user will generate the key for that file and send that key to the private cloud. In this way a user can upload the file into the public cloud.

File Downloading: When a user wants to download a particular file that he/she uploaded or any other user uploaded files on the public cloud. If he/she makes a request to the public cloud, it will provide the list of all files that many users uploaded on it. Among that, user selects one particular file form the list of the available files and enters the download option. In the meanwhile, private cloud will send a message that, enter the key for the file that is been generated by the user. Then the user enters the key for the file that is been generated. Private cloud will then checks for the key for that file and if that key is exact, that means the user is valid. Then only the user can actually download the file from the public cloud else ways user can't download the file. When user wants to download the file from the public cloud, it will be in the encrypted format then user can decrypt that file by using the same symmetric key.

V. SECURITY CONSIDERATIONS

This paper has addressed the problem of privacy preserving de-duplication and had proposed a new de-duplication system supporting for, the

Differential Authorization: Any authorized user will be able to access their individual token of his/her file to perform the duplicate check based on his privileges. Under this expectation, any unauthorized user cannot generate a token for the duplicate check, out of his access or without the support from the private cloud server.

Authorized duplicate check Any Authorized user will be able to utilize his/her individual private keys in order to generate a query for a certain file with the privileges he/she owned from the help of private cloud, the public cloud can perform duplicate check and can inform the user if there is any duplicate. The security considerations in this paper lie in two folds, including the security of file token and for the security of data files. For the security of file token, two characteristic are defined as unforgeability and indistinguishability of the token. The details are given below.

Unforgeability of file token/duplicate-check token: The user has to make registration in private cloud for generating file token. Utilizing the corresponding file token he/she can upload or download the files on public cloud. The users are not been allowed to plot with public cloud server to break the unforgeability of file tokens. In our system, the S-CSP is straightforward but curious and will be performing the duplicate check upon receiving the duplicate request from the users. The duplicate check token of users has to be published from the private cloud server in our scheme.

Indistinguishability of the file token/duplicate-check token: Any user without querying for some file token, to the private cloud server, he cannot get any effective information from the token, which includes the file information or the privilege information.

Data Confidentiality: Any unauthorized users without any suitable token, including the S-CSP and the private cloud server, should be prevented from accessing to the underlying plain text stored on S-CSP. Specifically, the goal of the adversary is, retrieving and recovering the file that doesn't belong to them. In our system, we have compared the previous definition of data confidentiality based on the convergent encryption; a higher level confidentiality is defined and attained.

VI. LITERATURE REVIEWS

The previous de-duplication systems cannot support differential authorization duplicate check, which is important in many of the applications [10]. To such an authorized de-duplication system, each user is issued with a set of privileges during the system initialization. The overview of cloud de-duplication is as follow:

POST-PROCESS DEDUPLICATION: In the post-process deduplication, the new data is been first stored on to the storage device and then only a process eventually will analyze the data looking for the duplication[13]. The main benefit is that there is no need to wait until the hash calculations and lookup to get completed before storing the data by this; we can ensure that store performance is not degraded. The implementations are offering policy-based operation that can give users the ability to defer optimization on "active" files, or to process the files based on type and location. The potential drawback is that you may unnecessarily have to store duplicate data for a moment, which is an issue if the storage system is nearly with full capacity [12].

IN-LINE DEDUPLICATION: In in-line de-duplication process the hash calculations are created on the target device. If the device finds a block that it is already stored on the system, it will not store the new block again, it will just references to the already existing block. The benefit of the in-line de-duplication when compared to post-process de-duplication is that, it requires less storage as data is not duplicated [14]. On the other side, it is been mostly argued that because of the hash calculations and lookups process takes so long, it mean that the data ingestion can be slower therefore reduce the backup throughput of the device[15]. Nevertheless, certain vendors with the in-line de-duplication have verified equipment with similar performance to their post-process de-duplication counterparts. The Post-process and the in-line de-duplication methods are often heavily debated.

SOURCE VERSUS TARGET DEDUPLICATION

There is another way to look upon; about the data de-duplication is by where it occurs [11]. That is when the de-duplication occurs closer to the data where it is been created; it is often referred to as "source de-duplication"[16]. When it occurs nearer to the data where it is stored, which is commonly called as the "target de-duplication." Source de-duplication makes sure that data on the data source is de-duplicated [17]. Generally this will take place directly within a file system. The file system will systematically scan new files by creating hashes and compare them to hashes of existing files.

VII. CONCLUSION

The authorized data de-duplication that was been proposed, to protect the data security that includes differential privileges of users during the duplicate check. In the new de-duplication constructions we have presented several supporting authorized duplicate check in the hybrid cloud architecture, in the duplicate-check tokens of files are generated by the private cloud server with the private keys. The Security consideration demonstrates that our arrangements are secure in terms of insider and outsider attacks specified from the proposed system security model.

REFERENCES

[1] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart "Dupless: Server-Aided Encryption for Deduplicated Storage", SEC'13 Proceedings of the 22nd USENIX conference on Security, ISBN: 978-1-931971-03-4, Pages 179-194, USENIX Association Berkeley, CA, USA ©**2013**.

[2] Paul Anderson and Le Zhang, "Fast and secure laptop backups with encrypted de-duplication", LISA'10 Proceedings of the 24th international conference on Large installation system administration, Article No. 1-8, USENIX Association Berkeley, CA, USA ©**2010**.

[3] Shweta Pochhi and Vanita Babanne,"A Survey on Secure and Authorized Data Deduplication", International Journal of Science and Research (IJSR),Volume 3 Issue 11, November **2014**.

[4] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, P. Lee, and Wenjing Lou "Secure deduplication with efficient and reliable convergent key management", IEEE Transactions on Parallel and Distributed Systems (TPDS), volume:25, Issue 6 **2014**.

[5] Shai Halevi, Danny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg, "Proofs of ownership in remote storage systems", CCS '11 Proceedings of the 18th ACM conference on Computer and communications security, ISBN: 978-1-4503-0948-6, Pages 491-500, ACM New York, NY, USA ©**2011**.

[6] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider, "Twin clouds: An architecture for secure cloud computing", In Workshop on Cryptography and Security in Clouds (WCSC 2011), **2011**.

[7] NIST's Policy on Hash Functions, Available: http://csrc.nist.gov/groups/ST/hash/policy.html [Online]., Sept. **2012**.

[8] D. Meister and A. Brinkmann, "Multi-Level Comparison of Data Deduplication in a Backup Scenario", SYSTOR '09 Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference, ISBN: 978-1-60558-623-6, Article No. 8, ACM New York, NY, USA ©**2009**.

[9] J.S. Plank and L. Xu, "Optimizing Cauchy Reed-Solomon Codes for Fault-Tolerant Network Storage Applications", in Proc. 5th IEEE Int'l Symp. NCA, Cambridge, MA, USA, July **2006**, pp.173-180.

[10] M.O. Rabin, "Fingerprinting by Random Polynomials", Center for Research in Computing Technology, Harvard University, Cambridge, MA, USA, Tech. Rep. TR-CSE-03-01, **1981**.

[11] M.O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing, Fault Tolerance", J. ACM, vol. 36, no. 2, pp. 335-348, Apr. **1989**.

[12] A. Rahumed, H.C.H. Chen, Y. Tang, P.P.C. Lee, and J.C.S. Lui, "A secure Cloud Backup System with Assured Deletion and Version Control", In Proc. 3rd Int'l Workshop Security Cloud Computing, pp. 160-167, **2011**.

[13] A.D. Santis and B. Masucci, "Multiple Ramp Schemes", IEEE Trans. Inf. Theory, vol. 45, no. 5, pp. 1720-1728, July **1999**.

[14] M.W. Storer, K. Greenan, D.D.E. Long, and E.L. Miller, "Secure Data Deduplication", in Proc. StorageSS, pp. 1-10, **2008**.

[15] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou "A Hybrid Cloud Approach for Secure Authorized De-duplication", in vol: pp no-99, IEEE, **2014**.

[16] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system", In ICDCS, pages 617–624, **2002**.

[17] D. Ferraiolo and R. Kuhn, "Role-based access controls", In 15th NIST-NCSC National Computer Security Conf., **1992**.

[18] V.P.Muthukumar and R.Saranya, "A Survey on Security Threats and Attacks in Cloud Computing", International Journal of Computer Sciences and Engineering, Page No : 120-125, Volume-02 , Issue-11, E-ISSN: 2347-2693, Nov - **2014**.

AUTHORS PROFILE

Suresh Jampana has done his Master of Computer Applications with first division and currently pursuing his M.Tech in Computer Science(CST), Department of Computer Science and System Engineering from Andhra University, Visakhapatnam. His areas of interest are Cloud Computing, Java Programming.

SMT. S.Jhansi Rani, M.Tech., Assistant Professor, Department of Computer Science and System Engineering, Andhra University, Visakhapatnam. Received M. Tech in Computer Networks(CN). Having 6 years of teaching experience. Interesting area is Computer Networks.