International Journal On Engineering Technology and Sciences – IJETS™
ISSN(P): 2349-3968, ISSN (O): 2349-3976
Volume III, Issue IV, April- 2016

AN INLINE HIGH PERFORMANCE DEDUPLICATION SYSTEM WITH SECURE AUTHORIZATION IN CLOUD STORAGE

AMRUTHA OC
M-TECH , DEPT OF CSE
Malabar Institute of Technology, Anjarakandy.

NAVYA RAMESH PP
ASSISTANT PROFESSOR, CSE DEPT
Malabar Institute of Technology, Anjarakandy.

Abstract

Data de duplication is one of the most important techniques. It is used as the data compression technique in the storage system to avoid the duplicate copies of the same data. In cloud storage it is used to reduce the amount of storage space. The de duplication techniques are applied to the inline scenarios. I-sieve is a de duplication system based on the open source code iSCSI. Design of corresponding index table and present a multi cache to reduce the RAM consumption and reduce the lookup performance.Convergent encryption techniques are used to encrypt the data before outsourcing to protect the confidentiality of the user sensitive data while supporting the deduplication. Diffrent from the previous de duplication systems(Traditional de duplication system) different privileges are assigned to the users. Also proposes several de duplication supporting authorized duplicate check in a hybrid cloud. To prevent the unauthorized access a secure proof of ownership protocol is used to provide the proof to the user

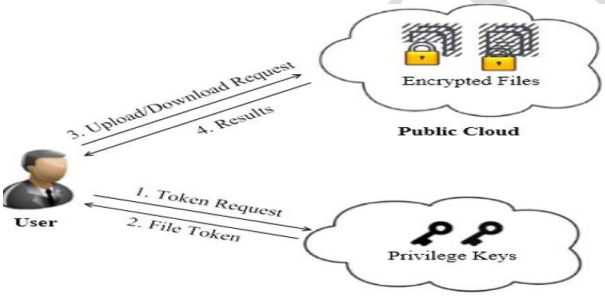
.Keywords— I-sieve; cloud storage; data deduplication, authorized duplicate check, confidentiality, hybrid cloud

1. Introduction

In the current trends ,the technologies have been advanced with the development of cloud computing and popularization of mobile internet, cloud services become more important in peoples life. With the development of information technology the large amount of data are stored in different storage system. To make the data management scalable a technique is used known as deduplication .Data deduplication is a data compression method is used to remove/eliminate different copies of same data in the storage system .It eliminate the redundant copies by keeping one copy of the data and pointing other redundant data to that copy. Data deduplication is one of the optimization technology to reduce the storage cost deduplication can work on many data types including block and sub-file , also known as super-block or fixed-size container file, whole-file, and hybrid-file. I-sieve is a backend storage module,and all the foreground clients access the storage service using the iSCSI protocol. Thus, I-sieve can be perceived as a data sieve,which eliminates much of the data redundancy in systems. The implementation of I-sieve is a module imbedded into iSCSI; therefore, it can be incorporated with many storage applications, .I-sieve consists of three key functional components, Deduplication engine, Multicache, and a Snapshot module. The datas uploaded into the cloud can be downloaded by the cloud users.The problem with this retrieval of the data will not provide secure authorization The encryption of the data before outsourcing into the cloud is time consuming. Encryption becomes more complex and critical in case of large amount of data. By using the data deduplication inside a hybrid cloud, the encryption will become simpler. Datas in the network,which is being shared by users and nodes in the network. Many large scale network uses the data cloud to store and share their data on the network. The node user, rights to upload or download data over the cloud. But many times diff erent user uploads the same data on the cloud. Which will create a duplication inside the cloud. If the user wants to retrieve or download the data from cloud, every time he has to use the two encrypted files of same data. The cloud will do same operation on the two copies of data files. Due to this process the data confidentiality and the security of the cloud get violated. It creates the workload on the operation of cloud.

To avoid this duplication of data and to maintain the confidentiality of the data in cloud we using the concept of Hybrid cloud.Hybrid cloud storage combines the advantages of scalability,reliability,rapid deployment and

potential cost savings of public cloud storage with the security and full control of private cloud storage. There are three entities dened in our system, that is, users, private cloud and S-CSP in public cloud as shown in Fig. The S-CSP performs deduplication by checking if the contents of two files are the same and stores only one of them. The access right to a file is defined based on a set of privileges. The notion of proof of ownership (PoW) enables users to prove their ownership of data copies to the storage server. Specically,PoW is implemented as an interactive algorithm(denotedbyPoW)run by a prover (i.e.,user)and a verifier(i.e.,storageserver). The verifier derives a short value (M) from a data copy M. To prove the ownership of the data copy M, the prover needs to send to the verifier such that = (M). An identication protocol can be described with two phases: Proof and Verify. In the stage of Proof, a prover/user U can demonstrate his identity to a verifier by performing some identification proof related to his identity.The input of the prover/user is his private key skU that is sensitive information such as private key of a public key in his certificate or credit card number etc. that he would not like to share with the other users. The verifier performs the verification with input of public information pkU related to skU. At the conclusion of the protocol the verifier outputs either accept or reject to denote whether the proof is passed or not.



2. Literature survey

Jibin Wang et.al[1]studied about the elimination of duplicate data in a storage system, commonly known as deduplication,is an eff ective technique to reduce storage costs. Thus, it has been applied to diff erent storage types, including backups, primary storage, within solid-state drives, and RAM. Although the general approach to deduplication is shared by all storage types, each poses specific challenge. The first contribution of this paper is a classification of deduplication systems according to six criteria that correspond to key design decisions techniques: granularity, locality, timing, indexing, technique, and scope. This classification identifies and describes the diff erent approaches used for each of them. This study proposes a novel deduplication architecture called I-sieve. The goal of I-sieve is to realize a high performance data sieve system based on iSCSI in the cloud storage system. We also design the corresponding index and mapping tables and present a multi-level cache using a solidstate drive to reduce RAM consumption and to optimize lookup performance. A prototype of I-sieve is implemented based on the open source iSCSI target, and many experiments have been conducted driven by virtual machine images and testing tools. The evaluation results show excellent deduplication and foreground performance. More importantly, I-sieve can co-exist with the existing deduplication systems as long as they support the iSCSI protocol.

Jin Li et.al[2] proposed method for aiming at efciently solving the problem of deduplication with diff erential privileges in cloud computing, we consider a hybrid cloud architecture consisting of a public cloud and a private cloud. Unlike existing data deduplication systems,the private cloud is involved as a proxy to allow data owner/users to securely perform duplicate check with diff erential privileges.Such an architecture is practical and has attracted much attention from researchers. The data owners only outsource their data storage by utilizing public cloud while the data operation is managed in private cloud. A new deduplication system supporting diff erential duplicate check is proposed under this hybrid cloud architecture where the S-CSP resides in the public cloud. The user is only allowed to perform the duplicate check for files marked with the corresponding privileges.In this paper presents an advanced scheme to support stronger security by encrypting with diff erential privilege keys.In this way,the users without corresponding privileges cannot perform the duplicate check. Furthermore, such unauthorized users cannot decrypt the ciphertext even collude with the S-CSP.

Austin T. Clements et.al[3] studied deduplication in the context of decentralized cluster file systems and also described a novel software system, DEDE, which provides block-level deduplication of a live, shared file system without any central coordination.

Furthermore, DEDE builds atop an existing file system without violating the file system's abstractions, allowing it to take advantage of regular file system block layout policies and in-place updates to unique data. Using our prototype implementation also plan to explore alternate indexing schemes that allow for greater control of deduplication policy. Additionally, they plan to further explore the trade-offs mentioned in this paper, such as block size versus metadata overhead, in-band versus out-of-band hashing, and sequential versus random index updates.DEDE represents just one of the many applications of deduplication to virtual machine environments

3. Proposed system

In the proposed system we are achieving the data deduplication by providing the proof of data by the data owner. This proof is used at the time of uploading of the file. Each file uploaded to the cloud is also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files. Before submitting duplicate check request for some file, the user needs to take this file and his own privileges as inputs. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched privilege stored in cloud. To support authorized deduplication, the tag of a file F will be determined by the file F and the privilege. To show the diff erence with traditional notation of tag, call it file token instead. To support authorized access, a secret key kp will be bounded with a privilege p to generate a file token. Let F,p = TagGen(F,kp) denote the token of F that is only allowed to access by user with privilege p. In another word,the token φ0 F,p could only be computed by the users with privilege p. As a result, if a file has been uploaded by a user with a duplicate token F,p, then a duplicate check sent from another user will be successful if and only if he also has the file F and privilege p. Such a token generation function could be easily implemented as H(F,kp), where H() denotes a cryptographic hash function. If a file duplicate is found, the user needs to run the PoW protocol POW with the S-CSP to prove the file ownership. If the proof is passed, the user will be provided a pointer for the file. Furthermore, a proof from the S-CSP will be returned, which could be a signature and a time stamp. The user sends the privilege set PF = pj for the file F as well as the proof to the private cloud server. Upon receiving the request, the private cloud server rst veries the proof from the S-CSP. If it is passed, the private cloud server computes F,pr = TagGen(F,kp) for all p satisfying R(p,pr) = 1 for each p PF-PU, which will be returned to the user. The user also uploads these tokens of the le F to the private cloud server. Then, the privilege set of the le is set to be the union of PF and the privilege sets defined by the other data owners.

Acronym	Description
S-CSP	Storage-cloud service provider
PoW	Proof of Ownership
(pk_U, sk_U)	User's public and secret key pair
k_F	Convergent encryption key for file F
P_U	Privilege set of a user U
P_F	Specified privilege set of a file F
$\phi'_{F,p}$	Token of file F with privilege p

Otherwise,if no duplicate is found,a proof from the S-CSP will be returned, which is also a signature on F,p, pkU and a time stamp. The user sends the privilege set PF = pj for the file F as well as the proof to the private cloud server. Upon receiving the request, the private cloud server rst veries the proof from the S-CSP. If it is passed, the private cloud server computes F,p = TagGen(F,kp) for all p satisfying R(p,p) = 1 and pPF. Finally, the user computes the encrypted file CF = EncCE(kF,F) with the convergent key kF =KeyGenCE(F) and uploads CF,F,pr with privilege PF.

A. *SymmetricEncyption*

Symmetric encryption uses a common secret key k to encrypt and decrypt the information. The key function of the scheme.

- *KeyGenSE : k* is the key generation algorithm that generates using security parameter 1λ.
- *KeyGenSE :* C is the symmetric encryption algorithm that takes the secret k and message M and then outputs the ciphertext C
- *DecSE(k,C):* M is the symmetric decryption algorithm that takes the secret and ciphertext C and then outputs the original message M.

B. *Convergent Encryption*

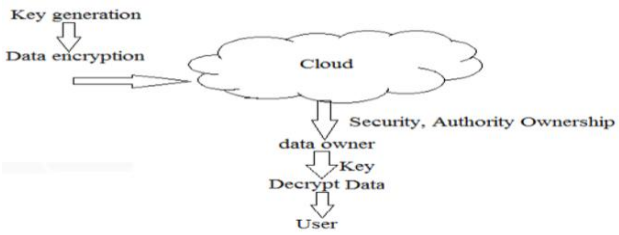
Convergent encryption provides data confidentiality in deduplication. A user (or data owner) derives a convergent key from each original data copy and encrypts the data copy with the convergent key. In addition, the user also derives a tag for the data copy, such that the tag will be used to detect duplicates.If two data copies are the same, then their tags are the same. To detect duplicates, the user first sends the tag to the server side to check if the identical copy has been already stored. Both of the convergent key and the tag are independently derived, and

the tag cannot be used to deduce the convergent key and compromise data confidentiality.The encrypted data copy and its corresponding tag will be stored on the server side.A convergent encryption scheme can be defined with the following four primitive functions

- *KeyGenCE(M)→K* is the key generation algorithm that maps a data copy M to a convergent key K;
- *EncCE(K,M)→C* is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a ciphertext C;
- *DecCE(K,C)→M* is the decryption algorithm that takes both the ciphertext C and the convergent key K as input s and the n outputs the original data copy M; and
- *TagGen(M)→T(M)*is the tag generation algorithm that maps the original data copy M and outputs a tag T(M).

C. Confidential Encryption

It provides data confidentiality in deduplication. A user derives a convergent key from each original data copy and encrypts the data copy with the convergent key. In addition, the user also derives a tag for the data copy, such that the tag will be used to detect duplicates.



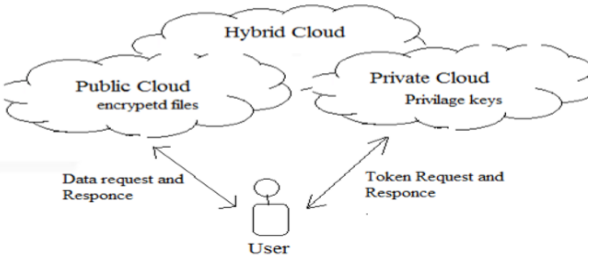
.

D. Proof of data

The user have to prove that the data which he want to upload or download is its own data. That means he have to provide the convergent key and verifying data to prove his ownership at server.

The notion of proof of ownership (PoW) enables users to prove their ownership of data copies to the storage server. Specically, PoW is implemented as an interactive algorithm (denoted by PoW) run by a prover (i.e., user)and a verifier. The verifier derives a short value φ(M) from a data copy M. To prove the ownership of the data copy M, the prover needs to send to the verfier such that φ0 = φ(M). The formal security definition for PoW roughly follows the threat model in a content distribution network, where an attacker does not know the entire file, but has accomplices who have the file.

IdenticationProtocol: An identification protocol π can be described with two phases: Proof and Verify. In the stage of Proof, a prover/user U can demonstrate his identity to a verifier by performing some identfication proof related to his identity.The input of the prover/user is his private key skU that is sensitive information such as private key of a public key in his certificate or credit card number etc. that he would not like to share with the other users. The verfiier performs the verification with input of public information pkU related to skU. At the conclusion of the protocol, the verifier outputs either accept or reject to denote whether the proof is passed or not



4. Conclusion

Here proposed a new deduplication engine supporting authorized duplicate check by using hybrid cloud architecture. In our system duplicate check tokens of files are generated with private keys by private cloud.The system provides authentication for each user for each file retrieval.The system provides privilages set to the users.The proposed System provide good storage utilization in cloud and provide security of the data over the cloud.

References

[1] Zhaogang Xu Hu Zhang Liang Li Jibin Wang, Zhigang Zhao and Ying Guo, I-sieve: An inline high performance deduplication system used in cloud storage, Tisnghua Sience and Tecchnology 20 (2015), no. 1, 17–27.

[2] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick PC Lee, and Wenjing Lou, A hybrid cloud approach for secure authorized deduplication, Parallel and Distributed Systems, IEEE Transactions on 26 (2015), no. 5, 1206–1216..

[3] A. T. Clements, I. Ahmad, M. Vilayannur, and J. Li, Decentralized deduplication in san cluster file systems, in Proceedings of the 2009 Conference on USENIX Annual Technical Conference, 2009.

[4] J. A. Paulo and J. Pereira, A survey and classification of storage deduplication systems, ACM Computing Surveys, vol. 47, no. 1, pp. 11: 1–11: 30, 2014.