ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Secure and Constant Cost Hybrid Cloud Storage Auditing With Deduplication

Divya Vijay Kandalkar
ME Final Year,
VLSI Design and Embedded System,
E&TC Engineering Dept,
S.V.I.T Nasik.

Prof. R.R.Bhambare
Associate Professor,
E&TC Engineering Dept,
S.V.I.T Nasik.

ABSTRACT:

Data deduplication is one of important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. To protect the confidentiality of sensitive data while supporting deduplication, the convergent encryption technique has been proposed to encrypt the data before outsourcing. To better protect data security, this paper makes the first attempt to formally address the problem of authorized data deduplication. Different from traditional deduplication systems, the differential privileges of users are further considered in duplicatecheck besides the data itself.We also present several new deduplication constructions supporting authorized duplicate check in a hybrid cloud architecture. Security analysis demonstrates that our scheme is secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement a prototype of our proposed authorized duplicate check scheme and conduct testbed experiments using our prototype. We show that our proposed authorized duplicate check scheme incurs minimal overhead compared to normal operations.

Index Terms:

Deduplication,cloud storage, authorized duplicate check, confidentiality, hybrid cloud.

I. Introduction:

Cloud computing provides seemingly unlimited "virtualized"resources to users as services across the wholeInternet, while hiding platform and implementation details.Today's cloud service providers offer both highlyavailable storage and massively parallel computing resources at relatively low costs.

As cloud computingbecomes prevalent, an increasing amount of data is beingstored in the cloud and shared by users with specifiedprivileges, which define the access rights of the storeddata. One critical challenge of cloud storage services isthe management of the ever-increasing volume of data.To make data management scalable in cloud computing,deduplication [11] has been a well-known techniqueand has attracted more and more attention recently.Data deduplication is a specialized data compressiontechnique for eliminating duplicate copies of repeatingdata in storage. The technique is used to improve storageutilization and can also be applied to network datatransfers to reduce the number of bytes that must besent. Instead of keeping multiple data copies with thesame content, deduplication eliminates redundant databy keeping only one physical copy and referring otherredundant data to that copy. Deduplication can takeplace at either the file level or the block level. For fileleveldeduplication, it eliminates duplicate copies of thesame file. Deduplication can also take place at the blocklevel, which eliminates duplicate blocks of data thatoccur in non-identical files.

In this paper, aiming at efficiently solving the problem ofdeduplication with differential privileges in cloud computing,we consider a hybrid cloud architecture consistingof a public cloud and a private cloud. Unlike existingdata deduplication systems, the private cloud is involvedas a proxy to allow data owner/users to securely performduplicate check with differential privileges. Such anarchitecture is practical and has attracted much attentionfrom researchers. The data owners only outsource theirdata storage by utilizing public cloud while the dataoperation is managed in private cloud. A new deduplicationsystem supporting differential duplicate checkis proposed under this hybrid cloud architecture wherethe S-CSP resides in the public cloud. The user is onlyallowed to perform the duplicate check for files markedwith the corresponding privileges.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Acronym	Description
S-CSP	Storage cloud service provider
PoW	Proof of ownership
(pk_U, sk_U)	User;s public key 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
$Ø'_{F,p}$	Token of file F with privilege P

Table 1: Notation in this paper

Furthermore, we enhance our system in security. Specifically, we present an advanced scheme to supportstronger security by encrypting the file with differentialprivilege keys. In this way, the users without correspondingprivileges cannot perform the duplicate check. Furthermore,such unauthorized users cannot decrypt theciphertext even collude with the S-CSP. Security analysis demonstrates that our system is secure in terms of thedefinitions specified in the proposed security model.Finally, we implement a prototype of the proposedauthorized duplicate check and conduct testbed experimentsto evaluate the overhead of the prototype. Weshow that the overhead is minimal compared to the normalconvergent encryption and file upload operations.

II. Architectural Diagram :

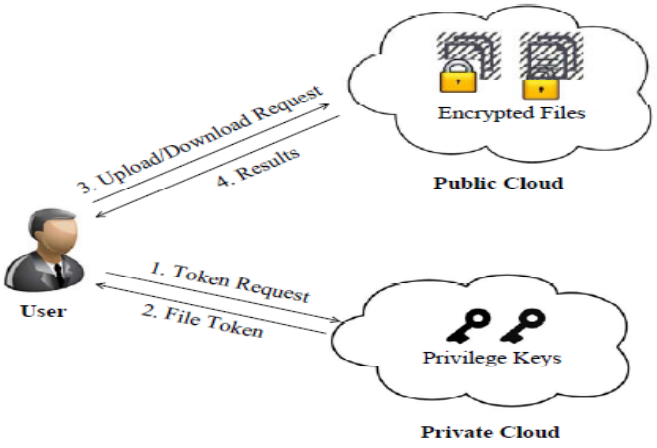

Figure 1. Architecture for Authorized Deduplication

A.System Model :

a. Hybrid Architecture for Secure Deduplication:

At a high level, our setting of interest is an enterprise-network, consisting of a group of affiliated clients (forexample, employees of a company) who will use theS-CSP and store data with deduplication technique. Inthis setting, deduplication can be frequently used inthese settings for data backup and disaster recoveryapplications while greatly reducing storage space. Suchsystems are widespread and are often more suitableto user file backup and synchronization applicationsthan richer storage abstractions. There are three entitiesdefined in our system, that is, users, private cloud andS-CSP in public cloud as shown in Figure 1. The S-CSPperforms deduplication by checking if the contents oftwo files are the same and stores only one of them.
The access right to a file is defined based on a setof privileges. The exact definition of a privilege variesacross applications. For example, we may define a rolebasedprivilege [9], [19] according to job positions (e.g.,Director, Project Lead, and Engineer), or we may definea time-based privilege that specifies a valid time period(e.g., 2014-01-01 to 2014-01-31) within which a file canbe accessed. A user, say Alice, may be assigned twoprivileges "Director" and "access right valid on 2014-01-01", so that she can access any file whose access roleis Director" and accessible time period covers 2014-01-01. Each privilege is represented in the form of a shortmessage called token. Each file is associated with somefile tokens, which denote the tag with specified privileges(see the definition of a tag in Section 2). A user computesand sends duplicate-check tokens to the public cloud forauthorized duplicate check.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

b.Securededuplication System:

To support authorized deduplication, the tagof a file F will be determined by the file F and the privilege.To show the difference with traditional notation oftag, we call it file token instead. To support authorizedaccess, a secret key kpwill be bounded with a privilege pto generate a file token. Let F;p= TagGen(F, kp) denotethe token of F that is only allowed to access by user withprivilege p. In another word, the token F;pcould onlybe computed by the users with privilege p. As a result, ifa file has been uploaded by a user with a duplicate tokenF;p, then a duplicate check sent from another user willbe successful if and only if he also has the file F andprivilegep. Such a token generation function could beeasily implemented as H(F, kp), where H(r) denotes acryptographic hash function.

B.Our Proposed System Description:

To solve the problems of the construction,we propose another advanced deduplication system supportingauthorized duplicate check. In this new deduplicationsystem, a hybrid cloud architecture is introducedto solve the problem. The private keys for privileges willnot be issued to users directly, which will be kept andmanaged by the private cloud server instead. In this way,the users cannot share these private keys of privilegesin this proposed construction, which means that it canprevent the privilege key sharing among users in theabove straightforward construction. To get a file token,the user needs to send a request to the private cloudserver. The intuition of this construction can be describedas follows.

a.System Setup. The privilege universe P is defined asin Section 4.1. A symmetric key kpifor each pi 2 Pwill be selected and the set of keys fkpigp Pwill besent to the private cloud. An identification protocol II= (Proof, Verify) is also defined, where Proof andVerify are the proof and verification algorithm respectively.Furthermore, each user U is assumed to have asecret key skUto perform the identification with servers.Assume that user U has the privilege set PU. It alsoinitializes a PoW protocol POW for the file ownershipproof. The private cloud server will maintain a tablewhich stores each user's public information pkUand itscorresponding privilege set PU. The file storage systemfor the storage server is set to be ?.

b.File Uploading. Suppose that a data owner wants toupload and share a file F with users whose privilege belongsto the set PF = fpjg. The data owner needs interactwith the private cloud before performing duplicate checkwith the S-CSP. More precisely, the data owner performsan identification to prove its identity with private keyskU. If it is passed, the private cloud server will findthe corresponding privileges PU of the user from itsstored table list. The user computes and sends the filetag F= TagGen(F) to the private cloud server, whowill return fF;pr= TagGen(F, kpr)gback to the userfor all prsatisfying R(p, pr) = 1 and p 2 PU. Then, theuser will interact and send the file token f F;prgto theS-CSP.

• If a file duplicate is found, the user needs to runthePoW protocol POW with the S-CSP to prove thefile ownership. If the proof is passed, the user willbe provided a pointer for the file. Furthermore, aproof from the S-CSP will be returned, which couldbe a signature on f F;prg, pkUand a time stamp.The user sends the privilege set PF = fpjgforthe file F as well as the proof to the private cloudserver. Upon receiving the request, the private cloudserver first verifies the proof from the S-CSP. If it ispassed, the private cloud server computes fF;pr=TagGen(F, kpr)g for all prsatisfying R(p, pr) = 1for each p 2 PF -PU, which will be returned to theuser. The user also uploads these tokens of the fileF to the private cloud server. Then, the privilegeset of the file is set to be the union of PF and theprivilege sets defined by the other data owners.

• Otherwise, if no duplicate is found, a proof fromthe S-CSP will be returned, which is also a signatureon fF;prg, pkUand a time stamp. Theuser sends the privilege set PF = fpjgfor thefileF as well as the proof to the private cloudserver. Upon receiving the request, the private cloudserver first verifies the proof from the S-CSP. If it ispassed, the private cloud server computes F;pr=TagGen(F, kpr)g for all prsatisfying R(p, pr) = 1and p 2 PF . Finally, the user computes the encryptedfile CF = EncCE(kF, F) with the convergentkey kF= KeyGenCE(F) and uploads fCF, fF;prggwith privilege PF .

c.File Retrieving. The user downloads his files in the sameway as the deduplication system in Section 4.1. That is,the user can recover the original file with the convergentkey kFafter receiving the encrypted data from the S-CSP.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

IIIResult analysis:

We implement cryptographic operations of hashing and encryption with the OpenSSL library [1].We also implementthe communication between the entities basedon HTTP, using GNU Libmicrohttpd [10] and libcurl [13].Thus, users can issue HTTP Post requests to the servers.Our implementation of the Clientprovides the followingfunction calls to support token generation anddeduplication along the file upload process.

• FileTag(File) - It computes SHA-1 hash of theFile as File Tag;

• TokenReq(Tag, UserID) - It requests the PrivateServer for File Token generation with the File Tagand User ID;

• DupCheckReq(Token) - It requests the StorageServer for Duplicate Check of the File by sendingthe file token received from private server;

• ShareTokenReq(Tag, {Priv.}) - It requests thePrivate Server to generate the Share File Token withthe File Tag and Target Sharing Privilege Set;

• FileEncrypt(File) - It encrypts the File withConvergent Encryption using 256-bit AES algorithmin cipher block chaining (CBC) mode, where theconvergent key is from SHA-256 Hashing of the file;And

• FileUploadReq(FileID, File, Token) – Ituploads the File Data to the Storage Server if the file is Unique and updates the File Token stored.

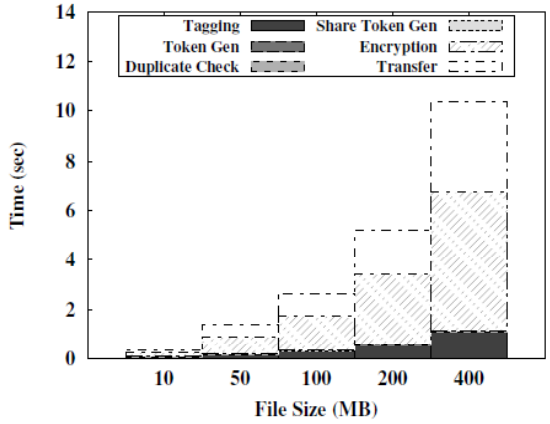

Figure 2. Time Breakdown for Different File Size

• TokenGen(Tag, UserID) - It loads the associated-privilege keys of the user and generate the tokenwith HMAC-SHA-1 algorithm; and

• ShareTokenGen(Tag, {Priv.}) - It generates the share token with the corresponding privilegekeys of the sharing privilege set with HMAC-SHA-1algorithm.Our implementation of the Storage Serverprovidesdeduplication and data storage with following handlersand maintains a map between existing files and associated-token with Hash Map.

• DupCheck(Token) - It searches the File to TokenMap for Duplicate; and

• FileStore(FileID, File, Token) - It storesthe File on Disk and updates the Mapping.

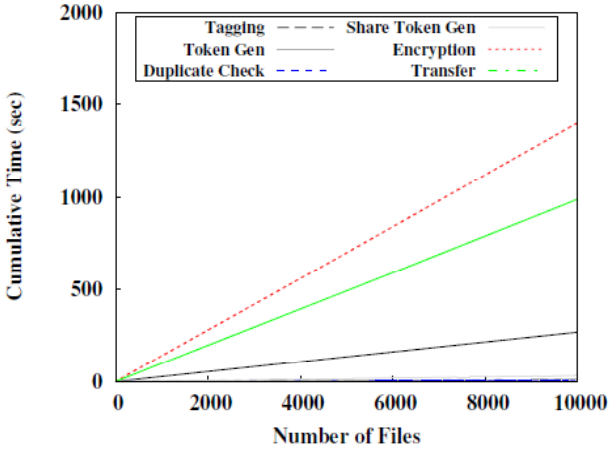

Figure 3. Time Breakdown for Different Number of StoredFiles

To evaluate the effect of number of stored files in the system,we upload 10000 10MB unique files to the systemand record the breakdown for every file upload. FromFigure 3, every step remains constant along the time.Token checking is done with a hash table and a linearsearch would be carried out in case of collision. Despiteof the possibility of a linear search, the time taken induplicate check remains stable due to the low collisionprobability.

A.Deduplication Ratio:

To evaluate the effect of the deduplication ratio, weprepare two unique data sets, each of which consists of50 100MB files.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

We first upload the first set as an initialupload. For the second upload, we pick a portion of 50files, according to the given deduplication ratio, from theinitial set as duplicate files and remaining files from thesecond set as unique files. The average time of uploadingthe second set is presented in Figure 4.

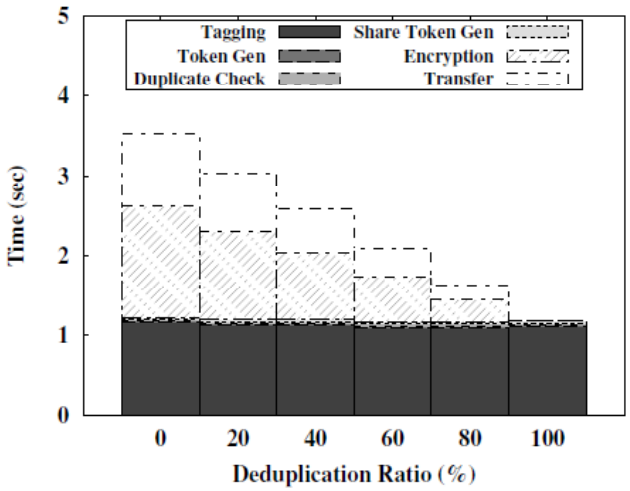

Figure 4. Time Breakdown for Different Deduplication Ratio

Acknowledgement:

We especially thanks to our department and our Institute for the great support regarding paper and their all views. I really thankful to my all staffs and my guide Mr. who showed me the way of successful journey of publishing paper and project work.

Conclusion:

In this paper, the notion of authorized data deduplicationwas proposed to protect the data security byincluding differential privileges of users in the duplicatecheck. We also presented several new deduplicationconstructions supporting authorized duplicate check inhybrid cloud architecture, in which the duplicate-checktokens of files are generated by the private cloud serverwith private keys. Security analysis demonstrates thatour schemes are secure in terms of insider and outsiderattacks specified in the proposed security model. As aproof of concept, we implemented a prototype of ourproposed authorized duplicate check scheme and conducttestbed experiments on our prototype. We showedthat our authorized duplicate check scheme incurs minimaloverhead compared to convergent encryption andnetwork transfer.

References:

[1] OpenSSL Project. http://www.openssl.org/.

[2] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.

[3] M. Bellare, S. Keelveedhi, and T. Ristenpart.Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart.Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296–312, 2013.

[5] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.

[6] M. Bellare and A. Palacio.Gq and schnorr identification schemes:Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.

[7] 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.

[8] 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.

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

[10] GNU Libmicrohttpd. http://www.gnu.org/software/libmicrohttpd/.

[11] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg.Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACM Conference on Computer and Communications Security, pages 491–500. ACM, 2011.