SAJET

South Asian Journal of Engineering and Technology Vol.2, No.17 (2016) 340–348

ISSN No: 2454-9614

An Efficient Security Based Authentication for Cloud Storage

P. Ashok kumar, S. Saradha*

Department of Computer Science and Engineering, Institute of Road and Transport Erode,Tamilnadu, India.

*Corresponding Author: P. Ashok Kumar

E-mail: ashokumar.myid@gmail.com

Received: 14/11/2015, Revised: 10/12/2015 and Accepted: 12/12/2016

Abstract

Data de duplication 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 de duplication, the convergent encryption technique has been proposed to encrypt the data before out sourcing .To better protect data security, this paper makes the first attempt to formally address the problem of authorized data de duplication. Different from traditional de duplication systems, the differential privileges of users are further considered in duplicate check besides the data itself. We also present several new de duplication constructions supporting authorized duplicate check in 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 test bed experiments using our prototype. We show that our proposed authorized duplicate check scheme incurs minimal over head compared to normal operations.

Keywords: De duplication, authorized duplicate check, confidentiality, hybrid cloud

Reviewed by* **ICETSET'16 *organizing committee*

1. Introduction

CLOUD computing provides seemingly unlimited ―virtualized‖ resources to users as services across threw hole Internet, while hiding platform and implementation details. Today's cloud service providers offer both high valuable storage and massively parallel computing resources at relatively low costs. As cloud computing becomes prevalent, an increasing amount of data is being stored in the cloud and shared by users with specified privileges, which define the access rights of the stored data. One critical challenge of cloud storage services is the management of the ever-increasing volume of data .To make data management scalable in cloud computing ,de duplication [17] has been a well-known technique and has attracted more and more attention recently. Data de duplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage .The technique is used to

improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies with the same content, de duplication eliminates redundant data by keeping only. One physical copy and referring other redundant data to that copy. De duplication can take place at either the file level or the block level. For file-level de duplication, it eliminates duplicate copies of the same file. De duplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files.

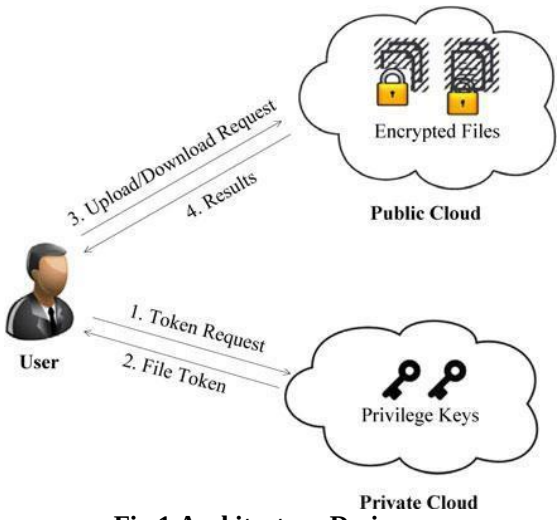

Fig 1 Architecture Design

1.1 Organization

The rest of this paper proceeds as follows. In Section 2, we briefly revisit some preliminaries of this paper. In Section 3, we propose the system model for our de duplication system .In Section 4, we propose a practical de duplication system with differential privileges in cloud computing. The security and efficiency analysis for the proposed system are respectively presented in Section 5. In Section 6, we present the implementation of our prototype, and in Section 7, we present test bed evaluation results. Finally we draw conclusion in Section 8.

1.2 Contributions

In this paper, aiming at efficiently solving the problem of de duplication with differential privileges in cloud computing, we consider a hybrid cloud architecture consisting of a public cloud and a private cloud. Unlike existing data de duplication systems, the private cloud is involved as a proxy to allow data owner/users to securely perform duplicate check with differential privileges. Such 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 de duplication system supporting differential 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 .Furthermore; we enhance our system in security. Specifically, we present an advanced scheme to support stronger security by encrypting the file with differential privilege keys. In this way, the users without corresponding privileges cannot perform the duplicate

check. Furthermore, such unauthorized users cannot decrypt the cipher text even collude with the S-CSP. Security analysis demonstrates the tour system is secure in terms of the definitions specified in the proposed security model .Finally; we implement a prototype of the proposed authorized duplicate check and conduct test bed experiments to evaluate the overhead of the prototype. We show that the overhead is minimal compared to the normal convergent.

2. Preliminaries

In this section, we first define the notations used in this paper, review some secure primitives used in our secure de duplication. The notations used in this paper are listed .Symmetric encryption. Symmetric encryption uses a common secret key k to encrypt and decrypt information. Asymmetric encryption scheme consists of three primitive functions:

_ KeyGenSEð1_Þ ! k is the key generation algorithm
that generates k using security parameter 1_;

_ Enc SE k; MÞ ! C is the symmetric encryption
algorithm that takes the secret k and message M and
then outputs the cipher text C; and

_ Dec SEk; CÞ ! M is the symmetric decryption algorithm that takes the secret k and cipher text C and then outputs the original message M.

Convergent encryption **[4], [8]** provides data confidentiality in de duplication. 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. Here, we assume that the tag correctness property **[4]** holds, i.e., 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. Note that both the convergent key and the tag are independently derived and the tag cannot be used to deduce the convergent key and compromise data confidentiality. Both the encrypted data copy and its corresponding tag will be stored on the server side. Formally, a convergent encryption scheme can be defined with four primitive functions:

_ Key GenCEðMÞ ! K is the key generation algorithm that maps a data copy M to a convergent key K;
Notations Used in This Paper
Li Er Al.: A Hybrid Cloud Approach for Secure Authorized Reduplication 1207
_ Encryption C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a cipher text C;
_ Decryption M is the decryption algorithm that takes both the cipher text C and the convergent key K as inputs and then outputs the original data copy M; and
_ Tag generation is the tag generation algorithm that maps the original data copy M and output **T (m)**

3. System Model

3.1. Hybrid Architecture for Secure De duplication

At a high level, our setting of interest is an enterprise network, consisting of a group of affiliated clients (for example, employees of a company) who will use the S-CSP and store data with de duplication technique. In this setting, de duplication can be frequently used in these settings for data backup and disaster recovery applications while greatly reducing storage space. Such systems are widespread and are often more suitable to user file backup and synchronization applications than richer storage abstractions. There are three entities defined in our system, that is, users, private cloud and S-CSP in public cloud as shown in Fig. 1. The S-CSP performs de duplication by checking if the contents of two files are the same and stores only one of them .Users have access to the private cloud server, a semi trusted third party which will aid in performing de duplicable encryption by generating file tokens for the requesting users. We will explain further the role of the private loud server below. Users are also provisioned with per user encryption keys and credentials (e.g., user certificates).

In this paper, we will only consider the file-level De duplication for simplicity. In another word, we refer a data copy to be a whole file and file-level de duplication which eliminates the storage of any redundant files. Actually, block-level de duplication can be easily de duplicated from file-level de duplication, which is similar to **[12].**

Specifically, to upload a file, a user first performs the file-level duplicate check. If the file is a duplicate, the null its blocks must be duplicates as well; otherwise, the user further performs the block-level duplicate check and identifies the unique blocks to be uploaded. Each data copy (i.e., a file or a block) is associated with a token for the duplicate check. S-CSP. This is an entity that provides a data storage service in public cloud. The S-CSP provides the data outsourcing service and stores data on behalf of the users. To reduce the storage cost, the S-CSP eliminates the storage of redundant data via de duplication and keeps only unique data. In this paper, we assume that S-CSP is always online and has abundant storage capacity and computation power.

3.2. Design Goals

In this paper, we address the problem of privacy-preserving de duplication in cloud computing and propose a new de duplication system supporting for _ Differential authorization. Each authorized user is able to get his/her individual token of his file to perform duplicate check based on his privileges. Under this assumption, any user cannot generate a token for duplicate check out of his privileges or without the aid from the private cloud server .Authorized duplicate check. Authorized user is able to use his/her individual private keys to generate query for certain file and the privileges he/she owned with the help of private cloud, while the public cloud performs duplicate check directly and tells the user if there is any duplicate .The security requirements considered in this paper lie in two folds, including the security of file token and security of data files. For the security of file token, two aspects are defined as un forget ability and in distinguish ability of file token.

4. Existing System

Before introducing our construction of differential de duplication, we present a straightforward attempt with the technique of token generation Tag Generation above to design such a de duplication system. The main idea of this basic construction is to issue corresponding privilege keys to each user, who will compute the file tokens and perform the duplicate check based on the privilege keys and files. In more details, suppose that there are N users in the system and the privileges in the universe is defined as P ¼fp1; . . . ; p. For each privilege p in P, a private key KP will be selected. For a user U with a set of privileges PU, he will be assigned the set of keys fkpigpi2PU.File uploading. Suppose that a data owner U with privilege set PU wants to upload and share a file F with users who have the privilege set PF ¼ f . The user computes and sends S-CSP the file token f0F;p ¼ Tag Generation for all p 2 PF. If a duplicate is found by the S-CSP, the user proceeds proof of ownership of this file with the S-CSP.

If the proof is passed, the user will be assigned a pointer, which allows him to access the file. Otherwise, if no duplicate is found, the user computes the encrypted file CF ¼ Enc CEk F; FÞ with the convergent key kF ¼ Key Generation and uploads, ff0F; pgÞ to the cloud server. The convergent key k F is stored by the user locally .File retrieving. Suppose a user wants to download a file F. It first sends a request and the file name to the S-CSP. Upon receiving the request and file name, the S-CSP will check whether the user is eligible to download F. If failed, the S-CSP sends back a signal to the user to indicate the download failure.

Otherwise, the S-CSP returns the corresponding cipher text CF. Upon receiving the encrypted data from the S-CSP, the user uses the key k F stored locally to recover the original file F.

4.1. Our Proposed System Description

To solve the problems of the construction in Section 4.1, we propose another advanced de duplication system supporting authorized duplicate check. In this new de duplication system, hybrid cloud architecture is introduced to solve 2the problem. The private keys for privileges will not be issued to users directly, which will be kept and managed by the private cloud server instead. In this way, the users cannot share these private keys of privileges in this proposed construction, which means that it can prevent the privilege key sharing among users in the above straightforward construction .To get a file token, the user needs to send a request to the private cloud server. The intuition of this construction can be described as follows. To perform the duplicate check for some file, the user needs to get the file token from the private cloud server. The private cloud server will also check the user's identity before issuing the corresponding file token to the user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file. Based on the results of duplicate check, the user either uploads this file or runs PoW.

Before giving our construction of the de duplication system, we define a binary relation R ¼ fðp; p0Þg as follows .Given two privileges p and p0, we say that p matches p0 if and only if Rðp; p0Þ ¼ 1. This kind of a generic binary relation definition could be instantiated based on the background of applications, such as the common hierarchical relation. More precisely, in a hierarchical relation, p matches p0 if p is a higher-level privilege. For example, in an enterprise management system, three hierarchical privilege levels are defined as Director, Project lead, and Engineer, where Director is at the top level and Engineer is at the bottom level. Obviously, in this simple

example, the privilege of Director matches the privileges of Project lead and Engineer. We provide the proposed de duplication system as follows. The privilege universe P is defined as in Section 4.1. A symmetric key k p i for each pi 2 P will be selected and the set of keys fkpigpi2P will be sent to the private cloud. An identification protocol P ¼ Proof; Verify Þ is also defined, where Proof and Verify are the proof and verification algorithm respectively. Furthermore, each user U is assumed to have a secret key s k U to perform the identification with servers. Assume that user U has the privilege set PU. It also initializes a PoW protocol POW for the file owner ship proof. The private cloud server will maintain a table which stores each user's public information p k U and its corresponding privilege set PU. The file storage system for the storage server

5. Security Analysis

Our system is designed to solve the differential privilege problem in secure de duplication. The security will be analyzed in terms of two aspects, that is, the authorization of duplicate check and the confidentiality of data. Some basic tools have been used to construct the secure de duplication, which are assumed to be secure. These basic tools include the convergent encryption scheme, symmetric encryption scheme, and the Pow scheme. Based on this assumption, we show that systems are secure with respect to the following.

5.1. Security of Duplicate-Check Token

We consider several types of privacy we need protect, that is, i) un forge ability of duplicate-check token: There are two types of adversaries, that is, external adversary and internal adversary. As shown below, the external adversary can be viewed as an internal adversary without any privilege. If a user has privilege p, it requires that the adversary cannot forge and output a valid duplicate token with any other privilege p0 on any file F, where p does not match p0.Furthermore, it also requires that if the adversary does not make a request of token with its own privilege from private cloud server, it cannot forge and output a valid duplicate token with p on any F that has been queried. The internal adversaries have more attack power than the external adversaries and thus we only need to consider the security against the internal attacker, ii) in distinguish ability of duplicate-check token: this property is also defined in terms of two aspects as the definition of un forget ability. First, if a user has privilege p, given a token f0, it requires that the adversary cannot distinguish which privilege or file in the token if p does not match p0.Furthermore, it also require that if the adversary does not make a request of token with its own privilege from private cloud server, it cannot distinguish a valid duplicate token with p on any other F that the adversary has not queried.

5.2 Confidentiality of Data

The data will be encrypted in our de duplication system before outsourcing to the S-CSP. Furthermore, two kinds of different encryption methods have been applied in our two constructions. Thus, we will analyze them respectively. In the scheme in Section 4.2, the data is encrypted with the traditional encryption scheme. The data encrypted with such encryption method cannot achieve semantic security as it is inherently subject to brute-force attacks that can recover files falling into a known set. Thus, several new security notations of privacy against

chosen-distribution attacks have been defined for unpredictable message. In another word, the adapted security definition guarantees that the encryptions of two unpredictable messages should be indistinguishable. Thus, the security of data in our first construction could be guaranteed under this security notion.

6. Implementation

We implement a prototype of the proposed authorized .De duplication system, in which we model three entities as separate C++ programs. A Client program is used to model the data users to carry out the file upload process. A Private Server program is used to model the private cloud which manages the private keys and handles the file token computation .A Storage Server program is used to model the S-CSP which stores and de duplicates files .We implement cryptographic operations of hashing and encryption with the Open SSL library **[1].** We also implement the communication between the entities based on HTTP, using GNU Lib micro http d **[10]** and libcurl **[13].**Thus, users can issue HTTP Post requests to the servers. Our implementation of the Client provides the following function calls to support token generation and de duplication along the file upload process.

_ File Tag(File)—It computes SHA-1 hash of the File as File Tag;

_ Token Req(Tag, User ID)—It requests the Private Server for File Token generation with the File Tag and User ID;

_ Dup Check Req (Token)—It requests the Storage Server for Duplicate Check of the File by sending the file token received from private server;

_ Share Token Req (Tag, {Priv.})—It requests the Private Server to generate the Share File Token with the File Tag and Target Sharing Privilege Set;

_ File Encrypt (File)—It encrypts the File with Convergent Encryption using 256-bit AES algorithm in cipher block chaining (CBC) mode, where the convergent key is from SHA-256 Hashing of the file;

File Upload Req (File ID, File, Token)—It uploads the File Data to the Storage Server if the file is Unique and updates the File Token stored. Our implementation of the Private Server includes corresponding request handlers for the token generation and maintains a key storage with Hash Map.

7. File Size

To evaluate the effect of file size to the time spent on different steps, we upload 100 unique files (i.e., without any de duplication opportunity) of particular file size and record the time break down. Using the unique files enables us to evaluate the worst-case scenario where we have to upload all file data. The average time of the steps from test sets of different file size .The time spent on tagging, encryption, upload increases linearly with the file size, since these operations involve the actual file data and incur file I/O with the whole file .In contrast, other steps such as token generation and duplicate check only use the file metadata for computation and therefore the time spent remains constant. With the file size increasing from 10 to 400 MB, the over head of the proposed authorization steps decreases from 14.9to 0.483 percent.

8. Related Work

Secure de duplication. With the advent of cloud computing, secure data de duplication has attracted much attention recently from research community. Yuan and Yu **[24]** proposed a de duplication system in the cloud storage to reduce the storage size of the tags for integrity check. To enhance the security of de duplication and protect the data confidentiality, Bellare et al. **[3]** showed how to protect. Time breakdown for different de duplication ratio. The data confidentiality by transforming the predictable message into unpredictable message.

Convergent encryption **[8]** ensures data privacy in de duplication. Bellare et al**. [4]** formalized this primitive as message-locked encryption, and explored its application in space-efficient secure outsourced storage. Xu et al. **[23]** also addressed the problem and showed a secure convergent encryption for efficient encryption, without considering issues of the key-management and block level de duplication. There are also several implementations of convergent implementations of different convergent encryption variants for secure de duplication (e.g., **[2], [18], [21], [22]).** It is known that some commercial cloud storage providers, such as Bit casa, also deploy convergent encryption.

9. Conclusion

In this paper, the notion of authorized data de duplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new de duplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer. And we can provide a security for images and multiple cloud are arranged and find the duplication.

References

[1] OpenSSL Project, (1998). [Online]. Available: http://www. openssl.org/

[2] P. Anderson and L. Zhang, ―Fast and secure laptop backups with encrypted de-duplication, ‖ in Proc. 24th Int. Conf. Large Installation Syst. Admin., 2010, pp. 29–40.

[3] M. Bellare, S. Keelveedhi, and T. Ristenpart, ―Dupless: Serve raided encryption for reduplicated storage, ‖ in Proc. 22nd USENIX Conf. Sec. Symp., 2013, pp. 179–194.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, ―Message-locked encryption and secure reduplications, ‖ in Proc. 32nd Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2013, pp. 296–312.

[5] M. Bellare, C. Namprempre, and G. Neven, ―Security proofs for identity-based identification and signature schemes,‖ J. Cryptol., vol. 22, no. 1, pp. 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 Proc. 22nd Annu. Int. Cryptol. Conf. Adv. Cryptol., 2002, pp. 162–177.

[7] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider, ―Twin clouds: An architecture for secure cloud computing,‖ in Proc. Workshop Cryptography Security Clouds, 2011, pp. 32–44.

[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 Proc. Int. Conf. Distrib. Comput. Syst., 2002, pp. 617–624.

[9] D. Ferraiolo and R. Kuhn, ―Role-based access controls, ‖ in Proc. 15th NIST-NCSC Nat. Comput. Security Conf., 1992, pp. 554–563.
[10] GNU Libmicrohttpd, (2012). [Online]. Available: 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 Proc. ACM Conf. Comput. Commun. Security, 2011, pp. 491–500.
[12] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, ―Secure reduplications with efficient and reliable convergent key management,‖ in Proc. IEEE Trans. Parallel Distrib. Syst., http:// doi.ieeecomputersociety.org/10.1109/TPDS.2013.284, 2013.
[13] libcurl, (1997). [Online]. Available: http://curl.haxx.se/libcurl/
[14] C. Ng and P. Lee, ―Revdedup: A reverse reduplications storage system optimized for reads to latest backups,‖ in Proc. 4th Asia- Pacific Workshop Syst., http://doi.acm.org/10.1145/2500727. 2500731, Apr. 2013.
[15] W. K. Ng, Y. Wen, and H. Zhu, ―Private data reduplications protocols in cloud storage,‖ in Proc. 27th Annu. ACM Symp. Appl. Comput., 2012, pp. 441–446.