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

A Cloud Approach for Secure Data Authorized Deduplication

M.Rajani
M.Tech Student,
Department of CSE,
Sree Rama institute of Technology and Science,
Kuppenakuntla, Penuballi, Khammam,TS India.

P.Nageswara Rao
Assistant Professor,
Department of CSE,
Sree Rama institute of Technology and Science,
Kuppenakuntla, Penuballi, Khammam,TS India.

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, authorized duplicate check, confidentiality, hybrid cloud.

INTRODUCTION:

CLOUD computing provides seemingly unlimited “virtualized” resources to users as services across the whole Internet, while hiding platform and implementation details. Today’s cloud service providers offer both highly available 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, deduplication [17] has been a well-known technique and has attracted more and more attention recently. Data deduplication 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, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Deduplication can take place at either the file level or the block level. For file-level deduplication, it eliminates duplicate copies of the same file. Deduplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files.

Existing System:

In the existing system, the system protecting the data confidentiality by transforming the predictable message into unpredictable message.Here another third party called key server is introduced to generate the file tag for duplicate check in which the data confidentiality is not more secured. Convergent encryption ensures data privacy in deduplication.The system is formalized this primitive as message-locked encryption, and explored its application in space-efficient secure outsourced storage in which the system will take more time duration to encrypt and decrypt.

DISADVANTAGES OF EXISTING SYSTEM:

The system will take more time to check data duplication in the cloud while using Convergent encryption techniques.Identical data copies of different users will lead to different cipher texts, making de duplication impossible. Data confidentiality will be loosed when third party key server is used for checking duplication.

Proposed System:

In this paper, aiming at efficiently solving the problem of deduplication with differential 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 differential 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 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.

ADVANTAGES OF PROPOSED SYSTEM:

The user is only allowed to perform the duplicate check for files marked with the corresponding privileges. We present an advanced scheme to support stronger security by encrypting the file with differential privilege keys. Reduce the storage size of the tags for integrity check. To enhance the security of deduplication and protect the data confidentiality,There is no any external key server to check the duplication, instead the duplication is checking in cloud server itself.

SYSTEM MODEL

Hybrid Architecture for Secure Deduplication:

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 deduplication technique.

In this setting, Deduplication 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 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 exact definition of a privilege varies across applications. For example, we may define a role-based privilege [9], [19] according to job positions (e.g., Director, Project Lead, and Engineer), or we may define a time-based privilege that specifies a valid time period (e.g., 2014-01- 01 to 2014-01-31) within which a file can be accessed.

A user, say Alice, may be assigned two privileges “Director” and “access right valid on 2014-01-01”, so that she can access any file whose access role is “Director” and accessible time period starts from 2014-01-01. Each privilege is represented in the form of a short message called token. Each file is associated with some file tokens, which denote the tag with specified privileges A user computes and sends duplicatecheck tokens to the public cloud for authorized duplicate check. Users have access to the private cloud server, a semitrusted third party which will aid in performing deduplicable encryption by generating file tokens for the requesting users. We will explain further the role of the private cloud server below.

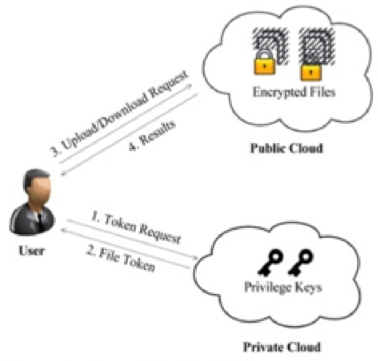

Fig. 1. Architecture for authorized deduplication.

Users are also provisioned with peruser encryption keys and credentials (e.g., user certificates). In this paper, we will only consider the file-level deduplication for simplicity. In another word, we refer a data copy to be a whole file and file-level Deduplication which eliminates the storage of any redundant files. Actually, block-level deduplication can be easily deduced from file-level deduplication, 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, then all 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.

Adversary Model:

Typically, we assume that the public cloud and private cloud are both “honest-but-curious”. Specifically they will follow our proposed protocol, but try to find out as much secret information as possible based on their possessions. Users would try to access data either within or out of the scopes of their privileges.

In this paper, we suppose that all the files are sensitive and needed to be fully protected against both public cloud and private cloud. Under the assumption, two kinds of adversaries are considered, that is, 1) external adversaries which aim to extract secret information as much as possible from both public cloud and private cloud; 2) internal adversaries who aim to obtain more information on the file from the public cloud and duplicate-check token information from the private cloud outside of their scopes. Such adversaries may include S-CSP, private cloud server and authorized users.

Design Goals:

In this paper, we address the problem of privacy-preserving deduplication in cloud computing and propose a new Deduplication 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 and 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.

SECURE DEDUPLICATION SYSTEMS:

Main idea. To support authorized deduplication, the tag of a file F will be determined by the file F and the privilege. To show the difference with traditional notation of tag, we 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 f0F;p ¼ TagGenðF; kpÞ denote the token of F that is only allowed to access by user with privilege p. In another word, the token f0F;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 f0F;p, then a duplicate check sentfrom 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 HðF; kpÞ, where Hð_Þ denotes a cryptographic hash function.

SECURITY ANALYSIS:

Our system is designed to solve the differential privilege problem in secure deduplication. 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 deduplication, which are assumed to be secure. These basic tools include the convergent encryption scheme, symmetric encryption scheme, and the PoW scheme.

IMPLEMENTATION:

We implement a prototype of the proposed authorized deduplication 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 deduplicates files. We implement cryptographic operations of hashing andencryption with the OpenSSL library [1].

We also implement the communication between the entities based on HTTP, using GNU Libmicrohttpd [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 deduplicationalong the file upload process.

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

FileTag(File)—It computes SHA-1 hash of the File as File Tag;TokenReq(Tag, UserID)—It requests the Private Server for File Token generation with the File Tag and User ID;
DupCheckReq(Token)—It requests the Storage Server for Duplicate Check of the File by sending the file token received from private server;
ShareTokenReq(Tag, {Priv.})—It requests the Private Server to generate the Share File Token with the File Tag and Target Sharing Privilege Set;
FileEncrypt(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; and FileUploadReq(FileID, 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.

EVALUATION:

We conduct testbed evaluation on our prototype. Our evaluation focuses on comparing the overhead induced by authorization steps, including file token generation and share token generation, against the convergent encryption and file upload steps. We evaluate the overhead by varying different factors, including 1) File Size, 2) Number of Stored Files, 3) Deduplication Ratio, 4) Privilege Set Size. We also evaluate the prototype with a real-world workload based on VM images.

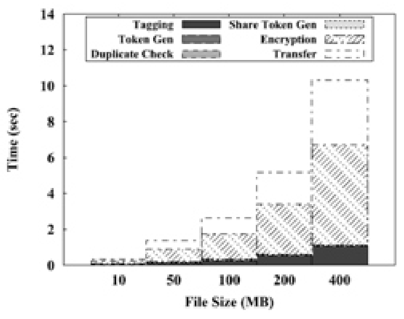

Fig. 2. Time breakdown for different file size.

We conduct the experiments with three machines equipped with an Intel Core-2-Quad 2.66 GHz Quad Core CPU, 4 GB RAM and installed with Ubuntu 12.04 32-Bit Operation System. The machines are connected with 1 Gbps Ethernet network.

We break down the upload process into six steps, 1) Tagging, 2) Token Generation, 3) Duplicate Check, 4) Share Token Generation, 5) Encryption, 6) Transfer. For each step, we record the start and end time of it and therefore obtain the breakdown of the total time spent. We present the average time taken in each data set in the figures.

RELATED WORK:

Secure deduplication. With the advent of cloud computing, secure data deduplication has attracted much attention recently from research community. Yuan and Yu [24] proposeda deduplication system in the cloud storage to reduce the storage size of the tags for integrity check. To enhance the security of deduplication and protect the data confidentiality, Bellare et al. [3] showed how to protect the data confidentiality by transforming the predictable message into unpredictable message. In their system, another third party called key server is introduced to generate the file tag for duplicate check. Stanek et al. [20] presented a novel encryption scheme that provides differential security for popular data and unpopular data. For popular data that are not particularly sensitive, the traditional conventional encryption is performed. Another twolayered encryption scheme with stronger security while supporting deduplication is proposed for unpopular data. In this way, they achieved better tradeoff between the efficiency and security of the outsourced data. Li et al. [12] addressed the key-management issue in block-level Deduplication by distributing these keys across multiple servers after encrypting the files.

CONCLUSIONS:

In this paper, the notion of authorized data Deduplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new deduplication 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 testbed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

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: Serveraided encryption for deduplicated 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 deduplication," 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 Deduplication 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 deduplication 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 deduplication protocols in cloud storage," in Proc. 27th Annu. ACM Symp. Appl. Comput., 2012, pp. 441–446.

[16] R. D. Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," in Proc. ACM Symp. Inf., Comput. Commun. Security, 2012, pp. 81–82.

[17] S. Quinlan and S. Dorward, "Venti: A new approach to archival storage," in Proc. 1st USENIX Conf. File Storage Technol., Jan. 2002, p. 7.

Author's:

M.Rajani is a student of Sree Rama Institute of Technology & Science, uppenakuntla,Penuballi, Khammam, TS,India.Presently she is Pursuing her M.Tech (CSE) from this college.Her area of interests includes Information Security, Cloud Computing, Data Communication & Networks.

Mr. P.Nageswara Rao is an efficient teacher, received M.Tech from JNTU Hyderabad is working as an Assistant Professor in Department of C.S.E, Sree Rama Institute of Technology & Science, Kuppenakuntla, Penuballi, Khammam, AP,India. He has published many papers in both National & International Journals. His area of Interest includes Data Communications & Networks, Database Management Systems, Computer Organization, C Programming and other advances in Computer Applications.