ISSN 2349-4506
Impact Factor: 2.265

Global Journal of Engineering Science and Research Management

A HYBRID CLOUD APPROACH FOR SECURE APPROVED DEDUPLICATION

Abhinandan K*, Prasanna Kumar M

*PG Student Department of Computer Science Shridevi Institute of Engineering and Technology Tumakuru, Karnataka, India

Asst Professor Department of Computer Science Shridevi Institute of Engineering and Technology Tumakuru, Karnataka, India

DOI: 10.5281/zenodo.50844

KEYWORDS: Deduplication, Endeavor, Testbed, Ciphertext.

ABSTRACT

Information Deduplication is one of imperative information pressure systems for wiping out copy duplicates of rehashing information, and has been generally utilized as a part of distributed storage to decrease the measure of storage room and spare data transmission. To secure the secrecy of touchy information while supporting Deduplication, the concurrent encryption system has been proposed to scramble the information before outsourcing. To better ensure information security, this methodology makes the main endeavor to formally address the issue of approved information Deduplication. Unique in relation to customary Deduplication frameworks, the differential benefits of clients are further considered in copy check other than the information itself. It likewise exhibit a few new Deduplication developments supporting approved copy check in a half and half cloud engineering. Security examination exhibits that our plan is secure as far as the definitions indicated in the proposed security model. As a proof of idea, we actualize a model of proposed approved copy check plan and direct testbed tests utilizing model. It demonstrates that proposed approved copy check plan brings about insignificant overhead contrasted with typical operations.

INTRODUCTION

To make data management scalable in cloud computing, deduplication 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 Although data deduplication brings a lot of benefits, security and privacy concerns arise as users' sensitive data are susceptible to both insider and outsider attacks .Traditional encryption, while providing data confidentiality ,is incompatible with data deduplication. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different ciphertexts, making deduplication impossible. Convergent encryption has been proposed to enforce data confidentiality while making deduplication feasible. It encrypts/decrypts a data copy with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the ciphertext to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies will generate the same convergent key and hence the same ciphertext . To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys. Thus, convergent encryption allows the cloud to perform deduplication on the ciphertexts and the proof of ownership prevents the unauthorized user to access the file.

OVERVIEW OF THE HYBRID CLOUD CONCEPTS HYBRID CLOUD

A hybrid cloud is a cloud computing environment in which an organization provides and manages some resources in-house and has others provided externally .For example, an organization might use a public cloud service, such

as Amazon Simple Storage Service(Amazon S3) for archived data but continue to maintain in house storage for operational customer data The concept of a hybrid cloud is meant to bridge the gap between high control, high cost "private cloud" and highly callable , flexible , low cost "public cloud".

"Private Cloud" is normally used to describe a VMware deployment in which the hardware and software of the environment is used and managed by a single entity.

The concept of a "Public cloud" usually involves some form of elastic/subscription based resource pools in a hosting provider datacenter that utilizes multi-tenancy. The term public cloud doesn't mean less security, but instead refers to multi-tenancy.

The concept revolves heavily around connectivity and data portability. The use cases are numerous: resource burst-ability for seasonal demand, development and testing on a uniform platform without consuming local resources, disaster recovery, and of course excess capacity to make better use of or free up local consumption.

VMware has a key tool for "hybrid cloud" use called "vCloud connector". It is a free plugin that allows the management of public and private clouds within the vSphere client. The tool offers users the ability to manage the console view, power status, and more from a "workloads" tab, and offers the ability to copy virtual machine templates to and from a remote public cloud offering.

HYBRID CLOUD 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. 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 rolebased privilege 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 covers

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. A user computes and sends duplicate-check 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. Users are also provisioned with per-user encryption keys and credentials.

Architecture for Authorized Deduplication:

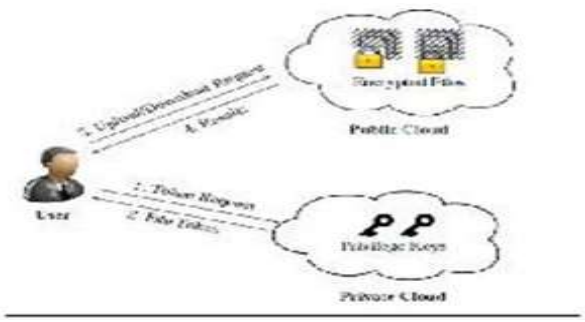

Fig 1: Architecture for Authorized deduplication

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, 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.

- 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 deduplication and keeps only unique data. In this paper, we assume that S-CSP is always online and has abundant storage capacity and computation power.
- Data Users. A user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users. In the authorized deduplication system, each user is issued a set of privileges in the setup of the system. Each file is protected with the convergent encryption key and privilege keys to realize the authorized deduplication with differential privileges.
- Private Cloud. Compared with the traditional deduplication architecture in cloud computing, this is a new entity introduced for facilitating user's secure usage of cloud service. Specifically, since the computing resources at data user/owner side are restricted and the public cloud is not fully trusted in practice, private cloud is able to provide data user/owner with an execution environment and infrastructure working as an interface between user and the public cloud. The private keys for the privileges are managed by the private cloud, who answers the file token requests from the users. The interface offered by the private cloud allows user to submit files and queries to be securely stored and computed respectively.

Notice that this is a novel architecture for data deduplication in cloud computing, which consists of a twin clouds (i.e., the public cloud and the private cloud). Actually, this hybrid cloud setting has attracted more and more attention recently. For example, an enterprise might use a public cloud service, such as Amazon S3, for archived data, but continue to maintain in-house storage for operational customer data.

Alternatively, the trusted private cloud could be a cluster of virtualized cryptographic co-processors, which are offered as a service by a third party and provide the necessary hardware based security features to implement a remote execution environment trusted by the 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.
- 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 unforgeability and indistinguishability of file token. The details are given below.

- Unforgeability of file token/duplicate-check token. Unauthorized users without appropriate privileges or file should be prevented from getting or generating the file tokens for duplicate check of any file stored

at the S-CSP. The users are not allowed to collude with the public cloud server to break the unforgeability of file tokens. In our system, the S-CSP is honest but curious and will honestly perform the duplicate check upon receiving the duplicate request from users. The duplicate check token of users should be issued from the private cloud server in our scheme.

- Indistinguishability of file token/duplicate-check token. It requires that any user without querying the private cloud server for some file token, he cannot get any useful information from the token, which includes the file information or the privilege information.
- Data confidentiality.Unauthorized users without appropriate privileges or files, including the S-CSP and the private cloud server, should be prevented from access to the underlying plaintext stored at S-CSP. In another word, the goal of the adversary is to retrieve and recover the files that do not belong to them. In our system, compared to the previous definition of data confidentiality based on convergent encryption, a higher level confidentiality is defined and achieved.

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 and encryption 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 deduplication along the file upload process.

- 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;
- 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.

- TokenGen(Tag, UserID)—It loads the associated privilege keys of the user and generate the token with HMAC-SHA-1 algorithm;
- ShareTokenGen(Tag, {Priv.})—It generates the share token with the corresponding privilege keys of the sharing privilege set with HMAC-SHA-1 algorithm.

Our implementation of the Storage Server provides deduplication and data storage with following handlers and maintains a map between existing files and associated token with Hash Map.

- DupCheck(Token)—It searches the File to Token Map for Duplicate.
- FileStore(FileID, File, Token)—It stores the File on Disk and updates the Mapping.

CONCLUSIONS

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 serve 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.

FUTURE SCOPE

It excludes the security problems that may arise in the practical deployment of the present model. Also, it increases the national security. It saves the memory by deduplicating the data and thus provide us with sufficient memory. It provides authorization to the private firms and protect the confidentiality of the important data.

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/libcur l/
14. C. Ng and P. Lee, "Revdedup: A reverse deduplication storage system optimized for reads to latest backups," in Proc. 4thAsiaPacificWorkshopSyst.,http://doi.acm.org/10.1145/250 0727. 2500731, Apr. 2013.