International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Realization of Hybrid Cloud Approach for Sheltered formal Deduplication

S. Srinivasareddy[1], M. Geethalatha[2]

[1]M. Tech, CS, Rise Krishna Sai Prakasam Group of Institutions

[2]Associate Professor, CS, Rise Krishna Sai Prakasam Group of Institutions

Abstract: ***One of important data compression techniques for eliminating duplicate copies of repeating data is Data deduplication, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. To protect the confidentiality 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 duplicate check besides the data itself. We also present several new deduplication 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 testbed experiments using our prototype. We show that our proposed authorized duplicate check scheme incurs minimal overhead compared to normal operations.***

Keywords: Deduplication, authorized duplicate check, confidentiality, hybrid cloud.

1. 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 ldata 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.

2. Literature Survey

In archival storage systems, there is a huge amount of duplicate data or redundant data, which occupy significant extra equipments and power consumptions, largely lowering down resources utilization (such as the network bandwidth and storage) and imposing extra burden on management as the scale increases. So data de-duplication, the goal of which is to minimize the duplicate data in the inter level, has been receiving broad attention both in academic and industry in recent years. In this paper, semantic data de-duplication (SDD) is proposed, which makes use of the semantic information in the I/O path (such as file type, file format, application hints and system metadata) of the archival files to direct the dividing a file into semantic chunks (SC). While the main goal of SDD is to maximally reduce the inter file level duplications, directly storing variable SCes into disks will result in a lot of fragments and involve a high percentage of random disk accesses, which is very inefficient. So an efficient data storage scheme is also designed and implemented: SCes are further packaged into fixed sized Objects, which are actually the storage units in the storage devices, so as to speed up the I/O performance as well as ease the data management. Primary experiments have demonstrated that SDD can further reduce the storage space compared with current methods .. With the advent of cloud computing, secure data deduplication has attracted much attention recently from research community. Yuan et al. proposed a 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. showed how to protect the data confidentiality by transforming the predicatable message into unpredicatable message. In their system, another third party called key server is introduced to generate the file tag for duplicate check. Stanek et al. presented a novel encryption scheme that provides the essential security for popular data and unpopular data. For popular data that are not particularly sensitive, the traditional conventional encryption is performed. Another two-layered encryption scheme with stronger security while supporting deduplication is proposed for unpopular data. In this way, they achieved better trade between the efficiency and security of the out-sourced data. Liet al. addressed the key management issue in block-level deduplication by distributing these keys across multiple servers after encrypting the files.

3. System Model

3.1 Secure Duplication with Hybrid Architecture:

By using the duplication technique, to store the data who will use S-CSP are consisted as group of affiliated client at high level. The main aim is enterprise all the network. To set the data back up and disaster recovery applications for reduce the storage space. We frequently go for de-duplication. Such systems are widespread and are often more suitable to user file backup and synchronization applications than richer storage abstractions.

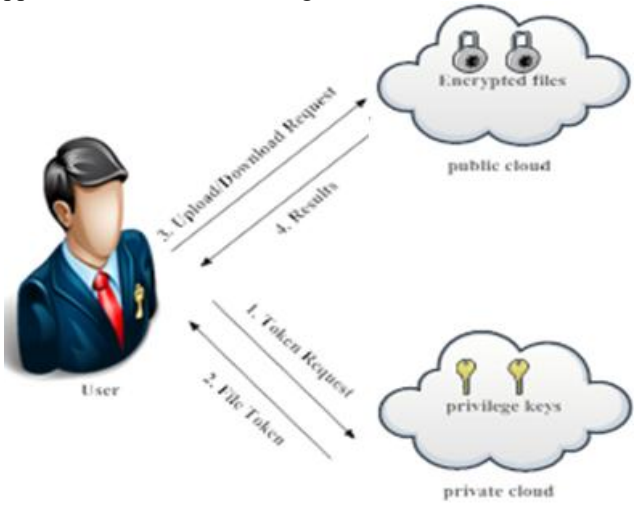

Figure 1: Working of authorized de-duplication There are three entities define in our system as shown in figure 1, those are,

- Users
- Private cloud
- S-CSP in public cloud

De-duplication performed by S-CSP by checking if the contents of two files are the same and stores only one of them. Based on the set of privileges, the access right of a file is defined. 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 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 privileges. A user computes and sends duplicate-check tokens to the public cloud for authorized 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 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.

- **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 de-duplication, 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 de-duplication 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 de-duplication with differential privileges.

- **Private Cloud:** Compared with the traditional de-duplication 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.

Hybrid clouds generally having twin clouds (private cloud and public cloud). This architecture is used for data de-duplication. 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.

3.2 Adversary Model

Both the public and private clouds are "honest-but-curious" try to find out as much secret information as possible based on their possessions either within or out of the limits of privileges users would try to access data. In this one, all the files are sensitive and needed to be fully protected against both public and private cloud. Assume two kinds of adversaries are considered.

- External adversaries which aim to extract secret information as much as possible from both public cloud and private cloud.
- 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.

These adversaries may include S-CSP, private cloud servers and authorized users.

3.3 Design Description

The detailed architecture of the design is showed in figure2. We can get the processing details from this architecture. Four different types of modules are present in the architecture. Data Owner Module, Encryption and Decryption Module, Remote User Module, Cloud Server Module. User login details are required to upload or download a file and the details of modules mentioned below,

a)Data Owner Module

- Data Owner login validations.
- Upload Files.
- Manipulates Encrypted files.
- Differential Authorization.

b)Encryption And Decryption Module

- Generate signs.
- Encrypts and uploads files.
- Decrypts and downloads files.
- Data confidentiality.

c) Remote User Module

- Accessing Files.
- Remote User login validations.

d)Cloud Server Module

- Authorized Duplicate Check.
- Accessing files.

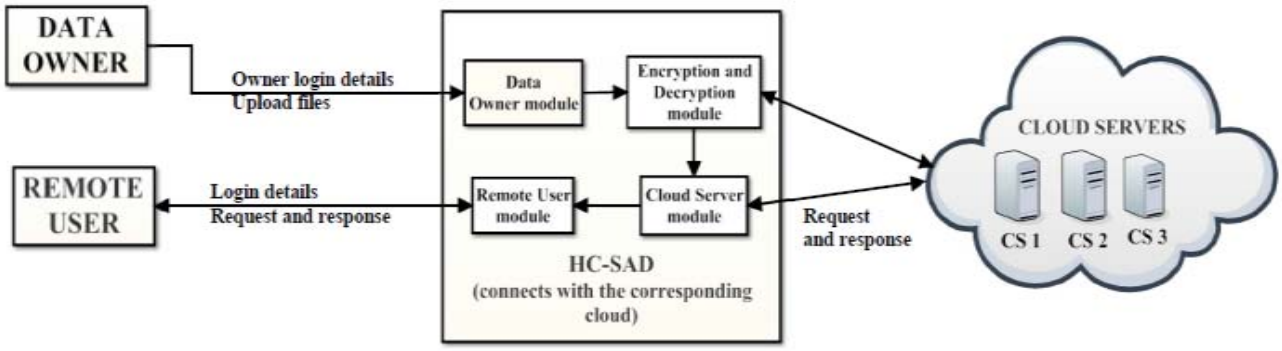

Figure 2: System Architecture design

3.3.1 New deduplication system

In this, we address the problem of privacy preserving de-duplication in cloud computing and propose a new deduplication system supporting for, the

- *Differential Authorization*: To perform duplicate check based on privilege of user is able to get his/her individual token. Without aid from the private cloud server and for the duplicate check outs token cannot generate by the user.
- *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-forge ability and in-distinguish ability 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.

4. 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 key 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 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 de-duplication 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; 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.
- *TokenGen(Tag, UserID)* - It loads the associated privilege keys of the user and generate the token with HMAC-SHA-1 algorithm; and
- *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 de-duplication 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; and
- *FileStore(FileID, File, Token)* - It stores the File on Disk and updates the Mapping.

5. Review

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 transefer.

6. Conclusion and Future Scope

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.

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] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou" A Hybrid Cloud Approach for Secure Authorized De-duplication" in vol: pp no-99, IEEE, 2014
[2] OpenSSL Project. http://www.openssl.org/.
[3] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In *Proc. of USENIX LISA*, 2010.
[4] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In *USENIX Security Symposium*, 2013.
[5] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure eduplication. In *EUROCRYPT*, pages 296– 312, 2013.
[6] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. *J. Cryptology*, 22(1):1–61, 2009.
[7] M. Bellare and A. Palacio. Gq and schnorr dentification schemes: Proofs of security against impersonation under active and concurrent attacks. In *CRYPTO*, pages 162–177, 2002.
[8] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. chneider. Twin clouds: An architecture for secure cloud computing. In *Workshop on Cryptography and Security in Clouds (WCSC 2011)*, 2011.

[9] 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.
[10] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15th NIST-NCSC National Computer Security Conf.*, 1992.
[11] GNU Libmicrohttpd. Http://www.gnu.org/software/libmicrohttpd/.
[12] 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.
[13] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure deduplication with efficient and reliable convergent key management. In *IEEE Transactions on Parallel and Distributed Systems*, 2013.
[14] libcurl. http://curl.haxx.se/libcurl/.
[15] C. Ng and P. Lee. Revdedup: A reverse deduplication storage system optimized for reads to latest backups. In *Proc. of APSYS*, Apr 2013.