International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Survey on Secure and Authorized Data Deduplication

Shweta Pochhi[1], Vanita Babanne[2]

[1, 2]Computer Engineering Department, RMD Sinhgad School of Engineering, Pune University, Pune

Abstract: ***Data deduplication looks for redundancy of sequences of bytes across very large comparison windows. Sequences of data (over 8 KB long) are compared to the history of other such sequences and it is ideal for highly redundant operations like backup which requires repeatedly copying and storing the same data set multiple times for recovery purpose. To protect the confidentiality of sensitive data while supporting deduplication, convergent encryption technique has been designed to encrypt the data. Convergent encryption enables duplicate files to coalesced into the space of a single file, even the files are encrypted with different users' keys. To overcome attacks, the notion of proofs-of-ownership (PoWs) has been introduced, which lets a client proficiently prove to a server that the client holds a file.***

Keywords: deduplication**,** Convergent Encryption, Proof of ownership, Authorized Duplicate Check, Differential Authorization

1. Introduction

Cloud computing provides unlimited "virtualized" resources to users as services across the whole Internet, while hiding platform and implementation details. Today's CSP(cloud service providers) offer both highly available storage and especially parallel computing resources at relatively low costs. As cloud computing becomes ubiquitous, an amount of data is being stored and shared by users with specified *privilege he cloud in t*, which define the access rights of the stored data. One significant challenge of cloud storage services is the management of the ever-increasing volume of data.

Cloud computing provides a low cost, scalable, location independent infrastructure for data management and storage. The rapid adoption of Cloud services is accompanied by increasing volumes of data stored at remote servers, hence techniques for saving disk space and network bandwidth are needed. A central up and coming concept in this context is deduplication, where the server stores a single copy of each file, in spite of of how many clients asked to store that file. All clients that store the file merely use links to the single copy of the file stored at the server. Moreover, if the server already has a copy of the file then clients do not even need to upload it again to the server, thus saving bandwidth as well as storage. In a typical storage system with deduplication, a client first sends to the server only a hash of the file and the server checks if that hash value already exists in its database. If the hash is not in the database then the server asks for the entire file. Otherwise, since the file already exists at the server (potentially uploaded by someone else) it tells the client that there is no need to send the file itself. Both way the server marks the client as an owner of that file, and from that point on there is no difference between the client and the original party who has uploaded the file. The client can therefore ask to restore the file, regardless of whether he was asked to upload the file or not.

Data deduplication is data compression technique for eliminating duplicate copies of repeating data in storage. This technique is used to improve storage utilization and can also be applied to network data transfers to decrease the number of bytes that must be sent. Deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy instead of keeping multiple data copies with the same content. Deduplication can take place at the file level and the block level. It eliminates duplicate copies of the same file at file level and also eliminates duplicate blocks of data that occur in non-identical files at the block level.

Data deduplication has certain benefits: Eliminating redundant data can extensively shrink storage requirements and improve bandwidth efficiency. Since primary storage has gotten cheaper over time, typically store many versions of the same information so that new workers can reuse previously done work. Some of the operations like backup store extremely redundant information.

Deduplication lowers storage costs as fewer disks are needed. It improves disaster recovery since there's far less data to transfer. Backup/archive data usually includes a lot of duplicate data. The similar data is stored over and over again, consuming unwanted storage space on disk or tape, electricity to power and cool the disk/tape drives and bandwidth for replication. This will create a chain of cost and resource inefficiencies within the organization.

While providing data confidentiality, traditional encryption is incompatible with data deduplication. Specifically, it requires different users to encrypt their data with their own keys. Thus, indistinguishable data copies of different users will lead to different cipher texts, making deduplication unfeasible. Convergent encryption has been proposed to impose data confidentiality while making deduplication feasible. It encrypts and decrypts a data copy with a *convergent key*, which is obtain by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users preserve the keys and send the ciphertext to the cloud. Because the encryption operation is deterministic and is derived from the data content, indistinguishable data copies will generate the same convergent key and hence the same ciphertext.

To avoid unauthorized access, a secure PoW (proof of ownership protocol) is also needed to provide the confirmation that the user indeed owns the same file when a duplicate is found. After the confirmation, consequent users with the same file will be provided a pointer from the server without needing to upload the similar file. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the equivalent data owners with their convergent keys. Thus, convergent encryption will allow the cloud to do deduplication on the ciphertexts and the proof of ownership (PoW) prevents the unauthorized user to access the file.

2. Related Work

2.1 Secure Deduplication

With the beginning of cloud computing, secure data deduplication has engrossed much attention recently from research community. Authors (Yuan et al.) proposed a deduplication system in the cloud storage to decrease the storage size of the tags for integrity check. To improve the security of deduplication and protect the data confidentiality, Bellare et al. showed how to protect the data confidentiality by transforming the expected message into random message.

In their system, another third party called key server has introduced to generate the file tag for duplicate check. Stanek et al. presented a new encryption scheme that provides differential security for accepted data and not accepted data. For popular data that are not mainly sensitive, the traditional conventional encryption is performed. Another two layered encryption scheme with stronger security while sustaining deduplication has been proposed for unpopular data.. Like this, they achieved better tradeoff between the efficiency and security of the outsourced data.

Li et al. addressed the key management concern in block level deduplication by distributing these keys across multiple servers after encrypting the files.

2.2 Convergent Encryption

Convergent encryption ensures data isolation in deduplication. Author Bellare et al. formalized this primitive as message locked encryption and explored its application in space efficient secure outsourced storage. Xu et al. also addressed the problem and showed a secure convergent encryption for efficient encryption without considering issues of the key management and block level deduplication.

To encrypt a file using convergent encryption, a client computes a cryptographically strong hash of the file content. The file is then encrypted using this hash value as a key. The hash value is then encrypted using the public keys of all authorized readers of the file and these encrypted values are attached to the file as metadata. Convergent encryption enables identical encrypted files to be recognized as identical, but there remains the problem of performing this identification across a large number of machines in a robust and decentralized manner.

There are also more than a few implementations of convergent implementations of different convergent encryption variants for secure deduplication. It is known that some commercial cloud storage providers also deploy convergent encryption.

2.3 Proof of ownership

Halevi et al. proposed the notion of PoW (“proofs of ownership”) for deduplication systems such that a client can efficiently confirm to the cloud storage server that he/she owns a file without uploading the file itself. Some PoW constructions based on the Merkle-Hash Tree are proposed to enable client-side deduplication which include the bounded leakage setting. Pietro and Sorniotti proposed another efficient PoW scheme by choosing the projection of a file onto some randomly selected bit positions as the file confirmation. Note that all the above schemes do not consider data privacy. Recently, Ng et al. extended PoW for encrypted files, but they do not address how to reduce the key management overhead.

Proof-of-ownership is a protocol in two parts between two players on a joint input F (which is the input file). First the verifier summarizes to itself the input file F and generates a (shorter) verification information v. Later on, the prover and verifier engage in an interactive protocol in which the prover has F and the verifier only has v, at the end of which the verifier either accepts or rejects. Hence a proof of-ownership is specified by a summary function S(_) (which could randomized and takes the input file F and a security parameter) and an interactive two-party protocol _(P $ V).

2.4 Twin Clouds Architecture

Bugiel et al. provided an architecture consisting of twin clouds for secure outsourcing of data and random computations to an untrusted commodity cloud. Zhang et al. also presented the hybrid cloud techniques to support privacy aware data intensive computing. To address the authorized deduplication problem over data in public cloud. The security model of systems is similar to those related work where the private cloud is assume to be sincere but curious.

3. Design Goals

Proposed deduplication system supporting for:

3.1 Differential Authorization

Every authorized user is able to get his/her individual token of his file to do duplicate check based on his privileges. Under this hypothesis, any user cannot generate a token for duplicate check out of his privileges or without the help from the private cloud server.

3.2 Authorized Duplicate Check

Authorized user is able to use his/her individual private keys to produce query for particular file and the privileges he/she owned with the aid of private cloud while the public cloud

performs duplicate check directly and tells the user if there is any duplicate. The security requirements 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.

a) **Unforgeability of file token/duplicate-check token**. Unauthorized users with inappropriate 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 get together with the public cloud server to break the unforgeability of file tokens. 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.

b) **Indistinguishability of file token/duplicate check token.** It requires that any user without querying the private cloud server for certain file token, he/she cannot get any valuable information from the token, which includes the file information or the privilege information.

c) Data Confidentiality. Unauthorized users without appropriate privileges or files including the S-CSP and the private cloud server should be prohibited from access to the primary plaintext stored at S-CSP. In other word, the goal of the adversary is to retrieve and recover the files that do not belong to them. Compared to the earlier definition of data confidentiality based on convergent encryption, a higher level confidentiality is defined and achieved.

4. System Description

To solve the problems of the construction, another advanced deduplication system supporting authorized duplicate has been proposed. In this new deduplication system, a hybrid cloud architecture has introduced to solve the problem. The private keys for privileges will not be issued to users directly which will be kept and managed by the private cloud server. 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 clear-cut construction. To get a file token, the user needs to send a request to the private cloud server. The perception of this construction can be described as follows.

To perform the duplicate check for certain file, the user needs to get the file token from the private cloud server. The private cloud server will check the user's identity before issuing the equivalent 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.

5. Conclusion

The idea of authorized data deduplication was proposed to protect the data security by including differential privileges of users in the duplicate check. 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 proposed schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a confirmation of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct testbed experiments on prototype. We proved that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

The large capacity of magnetic disks allows archival data to be retained and available on-line with performance that is similar to conventional disks. Stored on prototype server is over a decade of daily snapshots of two most important departmental file servers. These snapshots are stored in a small over 200 Gbytes of disk space. Today, 100 Gbytes drives cost less than $300 and IDE RAID controllers are included on many motherboards. A scaled down version of server could provide archival storage for a user at an attractive price. Tomorrow, when terabyte disks e had for the same price, it seems implausible that archival data will be deleted to reclaim space.

References

[1] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. *USENIX LISA*, 2010.
[2] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In *USENIX Security Symposium*, 2013.
[3] M. Bellare, S. Keelveedhi, T. Ristenpart. Message locked encryption and secure deduplication. In *EUROCRYPT*, pages 296–312, 2013.
[4] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon. In *ICDCS*, pages 617 -624, 2002. Reclaiming space from duplicate files in a serverless distributed file system.
[5] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou. Secure deduplication with efficient and reliable convergent key management. In *IEEE Transactions on Parallel and Distributed Systems* 2013.
[6] S. Quinlan and S. Dorward. Venti: a new approach to archival storage. *USENIX FAST*, Jan 2002.
[7] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui. A secure cloud backup system with assured deletion and version control. In *International Workshop on Security in Cloud Computing* 2011.
[8] Z. Wilcox-O'Hearn and B. Warner. Tahoe: the least-authority filesystem. *ACM StorageSS*, 2008.
[9] J. Xu, E.-C. Chang, and J. Zhou. In *ASIACCS*, pages 195–206, 2013.Weak leakage resilient client side deduplication of encrypted data in cloud storage.
[10] J. Yuan and S. Yu. Secure and constant cost public cloud storage auditing with deduplication 2013.