ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

2PDAS-Privacy Preserved Data Deduplication with Auto Attestation Strategy for Cloud Data

Mathumathi.B[1], Swapna.Mp[2]

Assistant Professor, Department of Computer Science, Sree Narayana Guru College, Coimbatore, Tamil Nadu, India[1]

M.Phil Scholar, Department of Computer Science, Sree Narayana Guru College, Coimbatore, Tamil Nadu, India[2]

Abstract: Data deduplication is the best data compression techniques for eliminating duplicate copies of same data and it is widely used to reduce the amount of storage space and save only single instance of a file. These deduplication mechanisms removes duplicate files from the storage cloud and replace them with the pointer of the original file. The idea of Deduplication optimizes the cost of data storage. Lower backup data means better RTO (recovery time objective). To protect the confidentiality of sensitive data while supporting deduplication, the privacy preserved data Deduplication with auto attestation strategy (2PDAS) has been proposed to protect the data before outsourcing. This study makes the first attempt to formally address the problem of authorized attestation based data Deduplication in order to maintain the data security. This 2PDAS is entirely different from traditional deduplication systems like ALG algorithm, the differential privileges of users are further considered in duplicate check along with the data which is deduplicated. We present a novel deduplication system which can be done by each user with the mechanism of Auto correction and Attestation based Integrity Verification system. The proposed system provides Guaranteed Data freshness and data management with privacy.

Keywords: Deduplication, privacy deduplicate check, confidentiality, cloud, integrity, redundancy.

I. INTRODUCTION

Data Deduplication is the well-known technique to make data management as secure and privacy in cloud computing and it is attracting more and more attention to ensure the security of cloud data. Data deduplication is a focused data compression technique for avoiding duplicate copies of repeating data in storage. The technique is used to make the storage as trustable and to improve storage utilization and it is also being applied to network data transfers to reduce the storage size.

To avoid keeping multiple data copies of the same content this deduplication eliminates duplicate data by keeping only one physical copy of the file and referring other duplicate data to that only one original copy. This deduplication can be either the file level or the block level. First type of file level deduplication eliminates duplicate copies of the same file. Second type called as block level eliminates duplicate blocks of data that occur in non-identical files. According to the original cloud storage schemes, some of schemes store the whole file into the storage server without any deduplication.

Thus, if there are two similar files, the cloud storage server would store redundant blocks between these two similar files. Therefore, the cloud storage capacity cannot be used properly. There are some cloud storage vendors using the technique of data deduplication when storing the uploaded files, the DropBox for example. Some data deduplication schemes calculate a hash value for each file used to check whether there is redundant hash value among uploaded files in the cloud storage.

Others translate a file into n blocks and then calculate a hash value to represent every block; therefore, the cloud storage server can examine the redundancy of every hash value of blocks.

Deduplication strategy can be categorized into two main strategies as follow, differentiated by the type of basic data units.

1). File-level deduplication: A file is a data unit when examining the data of duplication, and it typically uses the hash value of the file as its identifier. If two or more files have the same hash value, they are assumed to have the same contents and only one of these files will be stored.

2). Block-level deduplication: This strategy segments a file into several fixed-sized blocks or variable-sized blocks, and computes hash value for each block for examining the duplication blocks.

Though the existing method can find the duplicate blocks stored in the cloud storage server, it spends too much time on examining blocks which are duplicate. In this proposed system privacy preserved public data Deduplication based on auto attestation scheme is explained. Furthermore, a signature is computed for every uploaded file to ensure the integrity of file. Consider a cloud server that contains one hundred instances of the same one mega bytes of file attachment. Without data Deduplication if all the files are saved in the cloud server it requires one hundred mega bytes of storage space. If the data deduplication system is followed only one instance of the file is actually stored in the cloud server and all subsequent instances is just referenced back to the one saved copy of the data. This concept will reasonably reduce the storage and bandwidth demand to only one mega bytes.

II. LITERATURE REVIEW

Zhang, Yan, and Nayeem Ansari proposed in [1] major mechanisms involved in DRE techniques, including

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

fingerprinting, cache management, chunk matching, and decoding error recovery, have been discussed. For each mechanism, different approaches have been reviewed. In [2] presents LBFS, a network file system designed for low-bandwidth networks. LBFS exploits similarities between files or versions of the same file to save bandwidth. It avoids sending data over the network when the same data can already be found in the server's file system or the client's cache. [3] Describes a new idea called extreme Binning, for scalable and parallel deduplication, which is especially suited for workloads consisting of individual files with low locality. Existing approaches which require locality to ensure reasonable throughput perform poorly with such a workload. Extreme Binning exploits file similarity instead of locality to make only one disk access for chunk lookup per file instead of per chunk, thus alleviating the disk bottleneck problem. Eshghi, Kave, and Hsiu Khuern Tang given a new algorithm, TTTD, which performs much better than all the existing algorithms, and also puts an absolute size limit on chunk sizes. Using this algorithm can lead to a real improvement in the performance of applications that use content based chunking.

In [5] Zhu, Benjamin, Kai Li, and R. Hugo Patterson proposed three techniques employed in the production Data Domain deduplication file system to relieve the disk bottleneck. These techniques include: (1) the Summary Vector, a compact in-memory data structure for identifying new segments; (2) Stream-Informed Segment Layout, a data layout method to improve on-disk locality for sequentially accessed segments; and (3) Locality Preserved Caching, which maintains the locality of the fingerprints of duplicate segments to achieve high cache hit ratios.

[6] Provides an ADMAD: an Application-Driven Metadata Aware De-duplication Archival Storage System, which makes use of certain meta-data information of different levels in the I/O path to direct the file partitioning into more Meaningful data Chunks (MC) to maximally reduce the inter-file level duplications. Puzio, Pasquale, et al presented [7] ClouDedup, a secure and efficient storage service which assures block-level deduplication and data confidentiality at the same time. Although based on convergent encryption, ClouDedup remains secure thanks to the definition of a component that implement an additional encryption operation and an access control mechanism.

[8] Is proposed by Yuan, Jiawei, and Shucheng Yu and they provide a Proof of Retrievability (POR) and Proof of Data Possession (PDP) techniques assure data integrity for cloud storage. Proof of Ownership (POW) improves storage efficiency by securely removing unnecessarily duplicated data on the storage server. However, trivial combination of the two techniques, in order to achieve both data integrity and storage efficiency, results in non-trivial duplication of metadata (i.e., authentication tags), which contradicts the objectives of POW.

[9] Proposed a cryptographic primitive called proofs of ownership (PoW) to enhance security of client-side deduplication in cloud storage. In a proof of ownership scheme, any owner of the same file F can prove to the cloud storage that he/she owns file F in a robust and efficient way, in the bounded leakage setting where a certain amount of efficiently-extractable information about file F is leaked. Zhang, Kehuan, et al proposed in [10] Sedic, leverages the special features of MapReduce to automatically partition a computing job according to the security levels of the data it works on, and arrange the computation across a hybrid cloud. Specifically, we modified MapReduce's distributed file system to strategically replicate data, moving sanitized data blocks to the public cloud. Over this data placement, map tasks are carefully scheduled to outsource as much workload to the public cloud as possible, given sensitive data always stay on the private cloud.

III. PROPOSED SYSTEM

The proposed system is explained briefly in the architecture diagram. The proposed idea is focused on the public Deduplication system with data freshness guarantee. In the existing systems Deduplication concepts are done by the centralized authority so that there is no guarantee that the centralized authority is not compromised by any attackers. The main objective of the proposed system is to provide fresh data to the cloud users and at the same time the Deduplication process can also be done by the cloud users with the privacy using the attestation mechanism for each block.

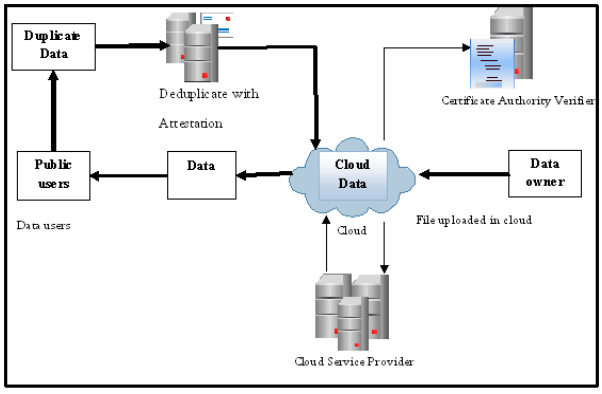

Fig 1: Architecture Diagram 2PDAS

2PDAS follows the Deduplication process with secure privacy based attestation scheme. During data upload and download process the Deduplication process will be taken and if the duplication processes done then the deduplicated file will be attested by the user signature for the verification process. This kind of attestation scheme avoids the attackers who target the data in the client machine and cloud server. The attestation scheme reveals if external users tried to duplicate the data.

Chunking Process

This process uses the file size filter to separate as tiny files. Intelligent chunker is used to break up the large size files into chunks. Backup files are divided into three categories the first type is compressed files, second type is static uncompressed files and the third type is dynamic uncompressed files. Static files cannot be editable and

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

dynamic files can be changed and the compressed files are chunked with file chunking mechanism. Uncompressed files are partitioned into fixed size chunks by Static Chunking process. Uncompressed files are broken into varied size chunks by content defined chunking process.

Importance of Deduplication:

Deduplication system provides two big advantages over a normal file system.

1. Reduced Storage Allocation - Deduplication can reduce storage needs by up to 90%-95% for files such virtual machine storage and main server backups.
2. Efficient Volume Replication – Basically unique data is stored in the disk the blocks which are need to be replicated. Depending on the application this can reduce traffic for replicating data by 90%-95%.

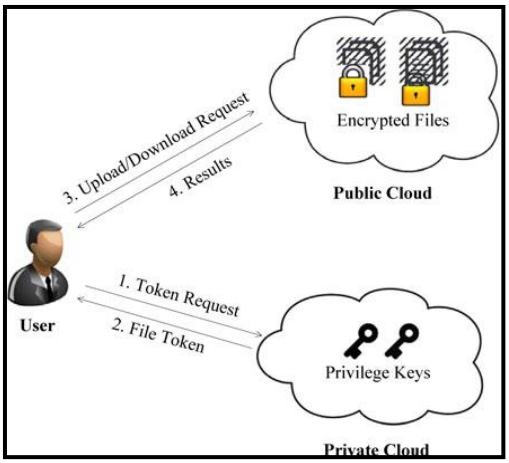

Fig 2: Public Deduplication Flow Diagram

2PDAS –ALGORITHM:

FileTagGeneration(File) : It computes SHA-1 hash of the File as File Tag.

TokenRequest(Tag, UserID) : It requests the Private Server for File Token generation with the File Tag and User ID.

DuplicateCheckRequest(Token) : It requests the Storage Server for Duplicate Check of the File by sending the file token received from private server.

ShareTokenRequest(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

FileUploadRequest(FileID, File, Token) : It uploads the File Data to the Storage Server if the file is Unique and updates the File Token stored.

Our proposed implementation 2PDAS provides Deduplication with low data storage with following parameter functions and maintains a map between existing files and associated token with Hash Map algorithm.

DEDUPLICATION FUNCTIONS:

The below fig 3: shows the processing time performance of the proposed system and it proves that it outperforms the earlier systems.

DuplicateCheck(Token) : It searches the File to Token Map for Duplicate; and

FileStoreOperation(FileID, File, Token) : It stores the File on Disk and updates the Mapping.

There are many Deduplication systems are have proposed and analyzed about the speed and accuracy. The following chart is implemented to compare with the existing Deduplication mechanisms. The following chart is deployed for all the Deduplication systems with respect to processing speed carried.

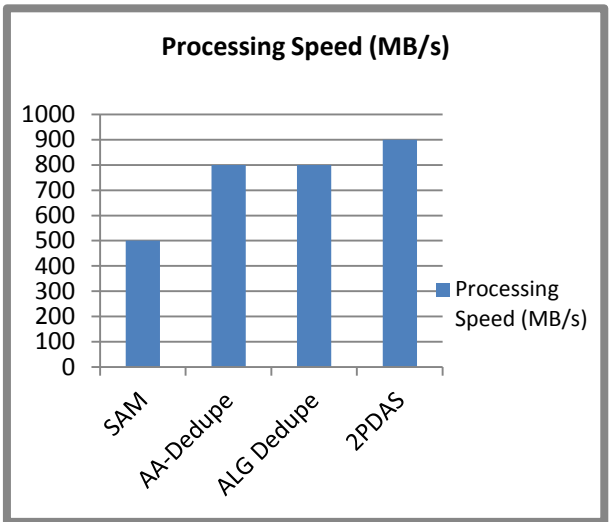

Fig 3: Processing Speed comparison chart

IV. CONCLUSION

The system proposed a new deduplication scheme called as 2PDAS-Privacy Preserved Data Deduplication with Auto Attestation strategy for cloud data provides a fast public Deduplication idea with attestation. The proposed idea split up the data into chunks and the deduplication efficiency of data chunking scheme among different applications differs greatly for data to data. After data is splitted into several blocks then blocks will be deduplicated by the public users with the attestation in both the local client and remote cloud server. It is proposed to protect the data security by including differential privileges of users in the duplicate check process with attestation. The authorized users of the cloud server can do the duplicate check with their privileges and once the Deduplication is done by any user then the credentials or secret key of the user who has done Deduplication is attested in the blocks of the data. Implemented Security analysis demonstrates that the proposed 2PDAS scheme is secure in terms of insider and outsider attacks who tried to attack the cloud server or data storage. The experiments showed that our privacy preserved deduplication system gives data freshness guarantee compared with the previous Deduplication systems.

REFERENCES

[1] Zhang, Yan, and Nayeem Ansari. "On protocol-independent data redundancy elimination." *Communications Surveys & Tutorials,*

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

IEEE 16.1 (2014): 455-472

[2] Muthitacharoen, Athicha, Benjie Chen, and David Mazieres. "A low bandwidth network file system." *ACM SIGOPS Operating Systems Review*. Vol. 35. No. 5. ACM, 2001.

[3] Bhagwat, Deepavali, et al. "Extreme binning: Scalable, parallel deduplication for chunk-based file backup." *Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on*. IEEE, 2009.

[4] Eshghi, Kave, and Hsiu Khuern Tang. "A framework for analyzing and improving content-based chunking algorithms." *Hewlett-Packard Labs Technical Report TR* 30 (2005): 2005.

[5] Zhu, Benjamin, Kai Li, and R. Hugo Patterson. "Avoiding the Disk Bottleneck in the Data Domain Deduplication File System." *Fast*. Vol. 8. 2008.

[6] Liu, Chuanyi, et al. "ADMAD: Application-Driven Metadata Aware De-duplication Archival Storage System." *Storage Network Architecture and Parallel I/Os, 2008. SNAPI'08. Fifth IEEE International Workshop on*. IEEE, 2008.

[7] Puzio, Pasquale, et al. "ClouDedup: secure deduplication with encrypted data for cloud storage." *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*. Vol. 1. IEEE, 2013.

[8] Yuan, Jiawei, and Shucheng Yu. "Secure and constant cost public cloud storage auditing with deduplication." *Communications and Network Security (CNS), 2013 IEEE Conference on*. IEEE, 2013.

[9] Xu, Jia, Ee-Chien Chang, and Jianying Zhou. "Weak leakage-resilient client-side deduplication of encrypted data in cloud storage." *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*. ACM, 2013.

[10] Zhang, Kehuan, et al. "Sedic: privacy-aware data intensive computing on hybrid clouds." *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011.