ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

A CONFINED DEDUPLICATION CLOUD SYSTEM WITH ENHANCED CREDIBILITY

Swetha.S [1], Matheswaran.V [2]

1. P.G. Student, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India
2. Asst.Professor, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India

ABSTRACT: For removing replication copies of data we use data deduplication process. As well as it is used in cloud storage to reduce memory space & upload bandwidth only one copy for each file stored in cloud that can be used by number of users. Deduplication process helps to improve storage reliability. One more challenge of privacy for sensitive data also arises when they are outsourced by users to cloud. The aim of this paper is to make the first attempt formalize the idea of distributed reliable deduplication system. In our proposed system we are going to develop new distributed deduplication systems which are highly reliable. In deduplication process data chunks are distributed across multiple cloud servers. The instead of using convergent encryption as in previous deduplication systems we use deterministic secret sharing scheme in distributed storage systems. So that we can achieve the required concepts for security that are data confidentiality and tag consistency. In the proposed security model, Security analysis demonstrates that our deduplication systems are secure.

KEYWORDS*:* Deduplication, reliability, distributed storage system, secret sharing, encryption

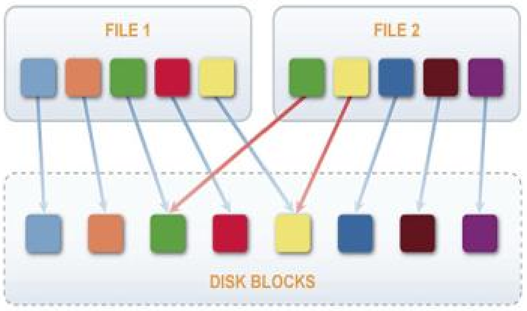
1.INTRODUCTION

Cloud computing enables companies to consume compute resources as a utility -- just like electricity -- rather than having to build and maintain computing infrastructures in-house. Cloud computing promises several attractive benefits for businesses and end users. Data Deduplication:Data is a method of reducing storage needs by eliminating redundant data. Only one unique instance of the data is actually retained on storage media, such as disk or tape. Redundant data is replaced with a pointer to the unique data copy. Data deduplication offers other benefits. Lower storage space requirements will save money on disk expenditures. The more efficient use of disk space also allows for longer disk retention periods, which provides better recovery time objectives for a longer time and reduces the need for tape backups. Data deduplication also reduces the data that must be sent across a WAN for remote backups, replication, and disaster recovery. In actual practice, data deduplication is often used in conjunction with other forms of data reduction such as conventional compression and delta differencing. Taken together, these three techniques can be very effective at optimizing the use of storage space.

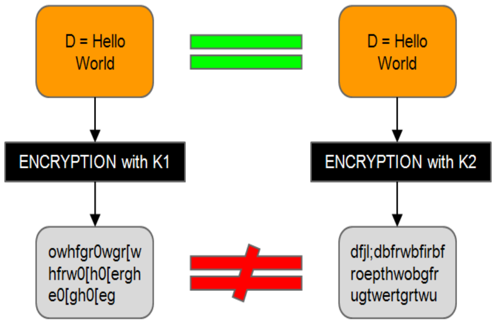
Data deduplication can generally operate at the file or block level. File deduplication eliminates duplicate files (as in the example above), but this is not a very

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

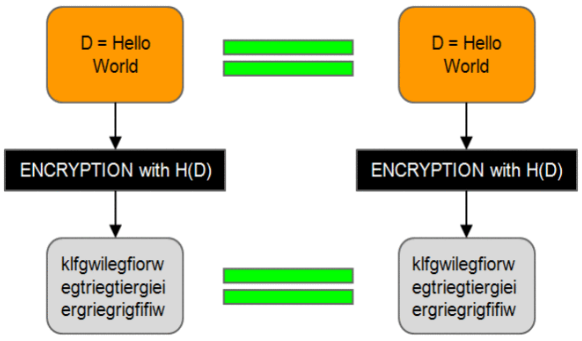
efficient means of deduplication. Block deduplication looks within a file and saves unique iterations of each block. Each chunk of data is processed using a hash algorithm such as MD5 or SHA-1. This process generates a unique number for each piece which is then stored in an index. If a file is updated, only the changed data is saved. That is, if only a few bytes of a document or presentation are changed, only the changed blocks are saved; the changes don't constitute an entirely new file. This behavior makes block deduplication far more efficient. However, block deduplication takes more processing power and uses a much larger index to track the individual pieces.Hash collisions are a potential problem with deduplication. When a piece of data receives a hash number, that number is then compared with the index of other existing hash numbers. If that hash number is already in the index, the piece of data is considered a duplicate and does not need to be stored again. Otherwise the new hash number is added to the index and the new data is stored.



From a high-level point of view, deduplication is very simple: **duplicate data is stored only once** and we keep pointers to the actual data.


D = Hello World
D = Hello World
ENCRYPTION with K1
ENCRYPTION with K2
owhfgr0wgr[w hfrw0[h0[ergh e0[gh0[eg
dfjl;dbfrwbfirbf roepthwobgfr ugtwertgrtwu

Deduplication is great but it doesn’t work on encrypted data because, somehow, it’s conflicting with encryption. Suppose we have two users who have the same data and want to store it remotely. Since they don’t trust the cloud provider, they will encrypt their data before uploading them to the cloud provider. Unfortunately, after the encryption, their data won’t be identical anymore and deduplication won’t possible! Indeed, the cloud provider cannot detect that those two data segments are identical.


D = Hello World
D = Hello World
ENCRYPTION with H(D)
ENCRYPTION with H(D)
klfgwilegfiorw egtriegtiergiei ergriegrigfifiw
klfgwilegfiorw egtriegtiergiei ergriegrigfifiw

There is a (simple) solution to this problem: Convergent Encryption! In convergent encryption, the encryption key is derived from the data (usually, it’s the hash of the data). This way, two different users will automatically use the same encryption key for the same data and thus generate the same ciphertext. At this point, the cloud storage provider

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

can still perform deduplication since he can detect duplicate data.

II.LITERATURE SURVEY

The Farsite distributed file system provides availability by replicating each file onto multiple desktop computers. Since this replication consumes significant storage space,[1] it is important to reclaim used space where possiblel. Measurement of over 500 desktop file systems shows that nearly half of all consumed space is occupied by duplicate files. We present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Our mechanism includes 1) convergent encryption, which enables duplicate files to coalesced into the space of a single file, even if the files are encrypted with different users' keys, and 2) SALAD, a SelfArranging, Lossy, Associative Database for aggregating file content and location information in a decentralized, scalable, fault-tolerant manner. Large-scale simulation experiments show that the duplicate-file coalescing system is scalable, highly effective, and fault-tolerant [4].

Cloud storage service providers such as Dropbox, Mozy, and others perform deduplication to save space by only storing one copy of each file uploaded [5]. Should clients conventionally encrypt their files, however, savings are lost. Message-locked encryption (the most prominent manifestation of which is convergent encryption) resolves this tension. However it is inherently subject to brute-force attacks that can recover files falling into a known set. We propose an architecture that provides secure deduplicated storage resisting brute-force attacks, and realize it in a system called DupLESS. In DupLESS, clients encrypt under message-based keys obtained from a key-server via an oblivious PRF protocol. It enables clients to store encrypted data with an existing service, have the service perform deduplication on their behalf, and yet achieves strong confidentiality guarantees. We show that encryption for deduplicated storage can achieve performance and space savings close to that of using the storage service with plaintext data [5].

An Information Dispersal Algorithm (IDA) is developed that breaks a file F of length L = (F (into n pieces F,, 1 5 i 5 n, each of length (F, 1 = L/m, so that every m pieces suffice for reconstructing F[9]. Dispersal and reconstruction are computationally efficient. The sum of the lengths (F, 1 is (n/m). L. Since n/m can be chosen to be close to I, the IDA is space eflicient. IDA has numerous applications to secure and reliable storage of information in computer networks and even on single disks, to fault-tolerant and efficient transmission of information in networks, and to communi-cations between processors in parallel computers. For the latter problem provably time-efftcient and highly fault-tolerant routing on the n-cube is achieved, using just constant size buffers. Categories and Subject Descriptors: E.4 [Coding and Information Theory]: nonsecret encoding schemes [9].

Data deduplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. Promising as it is, an arising challenge is to perform secure deduplication in cloud storage [11]. Christo Ananth et al. [2] discussed about a system, In this proposal, a neural network approach is proposed

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

for energy conservation routing in a wireless sensor network. Our designed neural network system has been successfully applied to our scheme of energy conservation. Neural network is applied to predict Most Significant Node and selecting the Group Head amongst the association of sensor nodes in the network. After having a precise prediction about Most Significant Node, we would like to expand our approach in future to different WSN power management techniques and observe the results. In this proposal, we used arbitrary data for our experiment purpose; it is also expected to generate a real time data for the experiment in future and also by using adhoc networks the energy level of the node can be maximized. The selection of Group Head is proposed using neural network with feed forward learning method. And the neural network found able to select a node amongst competing nodes as Group Head.

To this end, we propose Dekey , a new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers. Security analysis demonstrates that Dekey is secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs limited overhead in realistic environments [11].

Cloud storage systems are becoming increasingly popular. A promising technology that keeps their cost down is deduplication, which stores only a single copy of repeating data [12]. Client-side deduplication attempts to identify deduplication opportunities already at the client and save the bandwidth of uploading copies of existing files to the server. In this work we identify attacks that exploit client-side deduplication, allowing an attacker to gain access to arbitrary-size files of other users based on very small hash signatures of these files. More specifically, an attacker who knows the hash signature of a file can convince the storage service that it owns that file; hence the server lets the attacker download the entire file. (In parallel to our work, a subset of these attacks was recently introduced in the wild with respect to the Dropbox file synchronization service.) To overcome such attacks, we introduce the notion of proofs-ofownership (PoWs), which lets a client efficiently prove to a server that that the client holds a file, rather than just some short information about it. We formalize the concept of proof-of-ownership, under rigorous security definitions, and rigorous efficiency requirements of Petabyte scale storage systems. We then present solutions based on Merkle trees and specific encodings, and analyse their security. We implemented one variant of the scheme. Our performance measurements indicate that the scheme incurs only a small overhead compared to naive client-side deduplication [12].

III.EXISTING SYSTEM

A number of deduplication systems have been proposed based on various deduplication strategies such as client-side or server-side deduplications, file-level or block-level deduplications.Bellare et al. formalized this primitive as message-locked encryption, and explored its application in space efficient secure outsourced storage. There are also several implementations of convergent implementations of different convergent encryption variants for secure deduplication.Li addressed the key-management issue in block-level deduplication by distributing these keys across multiple servers after encrypting the files Bellare et al. showed how to protect data confidentiality by transforming the predictable message into a unpredictable message.[1]

Disadvantages of Existing System:

- Data reliability is actually a very critical issue in a deduplication storage system because there is only one copy for each file stored in the server shared by all the owners.
- Most of the previous deduplication systems have only been considered in a single-server setting.
- The traditional deduplication methods cannot be directly extended and applied in distributed and multi-server systems.

IV.PROPOSED ALGORITHM

System Architecture:

In this paper, we show how to design secure deduplication systems with higher reliability in cloud computing. We introduce the distributed cloud storage servers into deduplication systems to provide better fault tolerance. In more details, a file is first split and encoded into fragments by using the technique of secret sharing, instead of encryption mechanisms. These shares will be distributed across multiple independent storage servers. Furthermore, to support deduplication, a short cryptographic hash value of the content will also be computed and sent to each storage server as the fingerprint of the fragment stored at each server. Only the data owner who first uploads the data is required to compute and distribute such secret shares, while all following users who own the same data copy do not need to compute and store these shares any more. In other words, the secret shares of data will only be accessible by the authorized users who own the corresponding data copy.Four new secure deduplication systems are proposed to provide efficient deduplication with high reliability for file-level and block-level deduplication, respectively. The secret splitting technique, instead of traditional encryption methods, is utilized to protect data confidentiality. Specifically, data are split into fragments by using secure secret sharing schemes and stored at different servers [1].

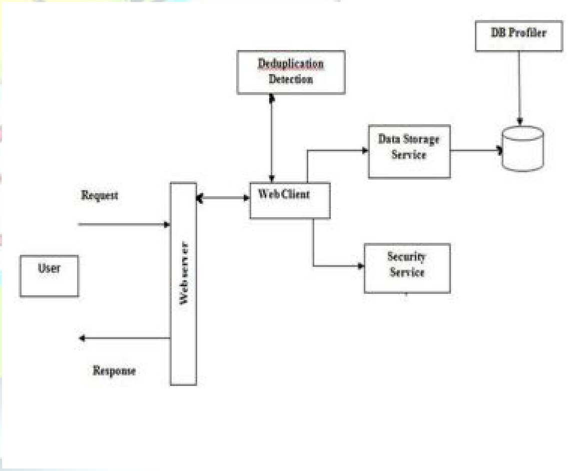


Figure 1: System Architecture of Secure deduplication

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

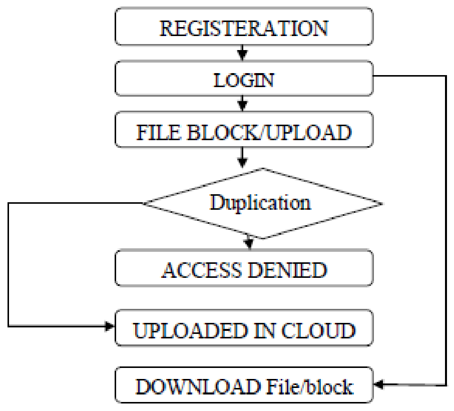


Figure 2: Data Flow Diagram of Secure deduplication

Modules:

System Model:

In this first module, we develop two entities: User and Secure-Cloud Service Provide. User: The 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. Furthermore, the fault tolerance is required by users in the system to provide higher reliability.

S-CSP:

The S-CSP is an entity that provides the outsourcing data storage service for the users. In the deduplication system, when users own and store the same content, the S-CSP will only store a single copy of these files and retain only unique data. A deduplication technique, on the other hand, can reduce the storage cost at the server side and save the upload bandwidth at the user side. For fault tolerance and confidentiality of data storage, we consider a quorum of S-CSPs, each being an independent entity. The user data is distributed across multiple S-CSPs.

Data Deduplication:

Data Deduplication involves finding and removing of duplicate data‟s without considering its fidelity.Here the goal is to store more data‟s with less bandwidth.Files are uploaded to the CSP and only the Data owners can view and download it. The Security requirements are also achieved by Secret Sharing Scheme.Secret Sharing Scheme uses two algorithms, share and recover. Data‟s are uploaded both file and block level and the finding duplication is also in the same process. This is made possible by finding duplicate chunks and maintaining a single copy of chunks.

File Level Deduplication Systems:

To support efficient duplicate check, tags for each file will be computed and are sent to S-CSPs.To upload a file F , the user interacts with S-CSPs to perform the deduplication.More precisely, the user firstly computes and sends the file tag ϕF = TagGen(F) to S-CSPs for the file duplicate check.If a duplicate is found the user computes and sendsit to a server via a secure channel.Otherwise if no duplicate is found the process continues,i.e secret sharing scheme runs and the user will upload a file to CSP. To download a file the user will use the secret shares and download it from the SCSP‟s .This approach provides fault tolerance and allows the user to remain accessible even if any limited subsets of storage servers fail.

Block Level Deduplication Systems:

In this module we will show to achieve fine grained block-level distributeddeduplication systems.In a block-level deduplication system, the user also needs to firstly perform the file-level deduplication before uploading his file.If no duplicate is found, the user divides this file into blocks and performs block-level deduplication.The System setup is similar to the file level deduplication except the parameter changes. To download a block the user gets the secret shares and downloads the blocks from CSP.

Advantage:

1. Distinguishing feature of our proposal is that data integrity, including tag consistency, can be achieved.

2. To our knowledge, no existing work on secure deduplication can properly address the reliability and tag consistency problem in distributed storage systems.

3. Our proposed constructions support both file-level and block-level deduplications.

4. Security analysis demonstrates that the proposed deduplication systems are secure in terms of the definitions specified in the proposed security model. In more details, confidentiality, reliability and integrity can be achieved in our proposed system. Two kinds of collusion attacks are considered in our solutions. These are the collusion attack on the data and the collusion attack against servers. In particular, the data remains secure even if the adversary controls a limited number of storage servers.

5. We implement our deduplication systems using the Ramp secret sharing scheme that enables high reliability and confidentiality levels. Our evaluation results demonstrate that the new proposed constructions are efficient and the redundancies are optimized and comparable with the other storage system supporting the same level of reliability.

V.CONCLUSION AND FUTURE WORK

We implement the secure distributed deduplication systems to improve the reliability of data while achieving the secret of the clients outsourced data. Four constructions were proposed to support file-level and fine-grained block-level data deduplication. The security of tag consistency and integrity were achieved. We implemented our deduplication systems using the Ramp secret sharing scheme and demonstrated that it incurs small encoding/decoding overhead compared to the network transmission overhead in regular upload/download operations.

REFERENCES

[1] H. Shacham and B. Waters, “Compact proofs of retrievability,” in ASIACRYPT, 2008, pp. 90–107.

[2] Christo Ananth, A.Nasrin Banu, M.Manju, S.Nilofer, S.Mageshwari, A.Peratchi Selvi, “Efficient Energy Management Routing in WSN”, International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Volume 1, Issue 1, August 2015,pp:16-19

[3]M. O. Rabin, “Fingerprinting by random polynomials,” Center for Research in Computing Technology, Harvard University, Tech. Rep. Tech. Report TR-CSE-03-01, 1981.

[4] 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, 2002, pp. 617–624.

[5] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: Serveraided encryption for deduplicated storage," in USENIX Security Symposium, 2013.

[6] "Message-locked encryption and secure deduplication," in EUROCRYPT, 2013, pp. 296–312.

[7] G. R. Blakley and C. Meadows, "Security of ramp schemes," in Advances in Cryptology: Proceedings of CRYPTO "84, ser. Lecture Notes in Computer Science, G. R. Blakley and D. Chaum, Eds. Springer-Verlag Berlin/Heidelberg, 1985, vol. 196, pp. 242–268.

[8] A. D. Santis and B. Masucci, "Multiple ramp schemes," IEEE Transactions on Information Theory, vol. 45, no. 5, pp. 1720–1728, Jul. 1999.

[9] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," Journal of the ACM, vol. 36, no. 2, pp. 335–

348, Apr. 1989.

[10] A. Shamir, "How to share a secret," Commun.ACM, vol. 22, no. 11, pp. 612–613, 1979.

[11] 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, 2014, pp. vol. 25(6), pp. 1615–1625.

[12] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems." in ACM Conference on Computer and Communications Security, Y. Chen, G. Danezis, and V. Shmatikov, Eds. ACM, 2011, pp. 491–500.

[13] J. S. Plank, S. Simmerman, and C. D. Schuman, "Jerasure: A library in C/C++ facilitating erasure coding for storage applications - Version 1.2," University of Tennessee, Tech. Rep. CS-08-627, August 2008.

[14] J. S. Plank and L. Xu, "Optimizing Cauchy Reed-solomon Codes for fault-tolerant network storage applications," in NCA-06: 5th IEEE International Symposium on Network Computing Applications, Cambridge, MA, July 2006.

[15] C. Liu, Y. Gu, L. Sun, B. Yan, and D. Wang, "R-admad: High reliability provision for large-scale de-duplication archival storage systems," in Proceedings of the 23rd international conference on Supercomputing, pp. 370–379.

[16] M. Li, C. Qin, P. P. C. Lee, and J. Li, "Convergent dispersal: Toward storage-efficient security in a cloud-of-clouds," in The 6th USENIX Workshop on Hot Topics in Storage and File Systems, 2014.

[17] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de-duplication," in Proc. of USENIX LISA, 2010.

[18] Z. Wilcox-O"Hearn and B. Warner, "Tahoe: the least-authority filesystem," in Proc. of ACM StorageSS, 2008.

[19] 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 3rd International Workshop on Security in Cloud

Computing, 2011.

[20] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, "Secure data deduplication," in Proc. of StorageSS, 2008.

[21] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage," in Technical Report, 2013.

[22] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services: Deduplication in cloud storage." IEEE Security & Privacy, vol. 8, no. 6, pp. 40–47, 2010.

[23] R. D. Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication." in ACM Symposium on Information, Computer and Communications Security, H. Y. Youm and Y. Won, Eds. ACM, 2012, pp. 81–82.

[24] J. Xu, E.-C.Chang, and J. Zhou, "Weak leakage-resilient client-side deduplication of encrypted data in cloud storage," in ASIACCS, 2013, pp. 195–206.

[25] W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage." in Proceedings of the 27th Annual ACM Symposium on Applied Computing, S. Ossowski and P. Lecca, Eds. ACM, 2012, pp. 441–446.

[26]G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS "07. New York, NY, USA: ACM, 2007, pp. 598–609. [Online].Available:http://doi.acm.org/10.1145/13152 45.1315318

[27]A. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS "07. New York, NY, USA: ACM, 2007, pp. 584–597. [Online].Available:http://doi.acm.org/10.1145/13152 45.1315317