IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 06-12
www.iosrjournals.org

Ensuring security for Fixed Block Level Deduplication in Cloud Backup

Akhila K[1], Amal Ganesh[2], Sunitha C[3]
(PG Student, VAST, Thrissur)

Abstract: *In the digital world, data is of prime importance for individuals as well as for organizations. As the amount of data being generated increases exponentially with time, duplicate data contents being stored cannot be tolerated. Thus, employing storage optimization techniques is an essential requirement to large storage areas like Cloud. Deduplication is one such storage optimization technique that avoids storing duplicate copies of data. Deduplication over encrypted data brings about more effective storage optimization in large storage areas like Cloud, since such storage areas are resident more of encrypted data. Data deduplication techniques used in large storage areas are widely applied over unencrypted data at file level. Some storage services even provide file level deduplication over encrypted data. Thus, there is a need for having more granular level of deduplication over encrypted data, such as block level deduplication. Objective of this project is to implement fixed block level deduplication over encrypted data with block keys using Cloud and backup services*.
Keywords: *Deduplication; Convergent encryption; Cloud storage; Proof of Ownership*

I. Introduction

With numerous benefits of cloud storage such as cost savings, accessibility, scalability etc., users around the world tend to shift their invaluable data to cloud storage. As the data generation rates are increasing, it is a tedious task for cloud storage providers to provide efficient storage. Cloud storage providers uses different techniques to improve storage efficiency and one of leading technique employed by them is deduplication, which claims to be saving 90 to 95% of storage [1],[2]. Data Deduplication technique evolved as an simple storage optimization technique in secondary then widely adapted in primary storage as well as larger storage areas like cloud storage area. Now, data deduplication is widely used by various cloud storage providers like Dropbox [3], Amazon S3 [4], Google Drive [5],etc. Data once deployed to cloud servers, its beyond the security premises of the data owner, thus most of them prefer to outsource their in an encrypted format. Data encryption by data owners eliminates cloud service providers chance of deduplicating it since encryption and deduplication techniques have conflicting strategies, i.e., data encryption with a key converts data into an unidentifiable format called cipher text thus encrypting, even the same data, with different keys may result in different cipher texts , making deduplication less feasible .However, performing encryption is essential to make data secure, at the same time, performing deduplication is essential for achieving optimized storage. Therefore, deduplication and encryption need to work in hand to hand to ensure secure and optimized storage. Various techniques and approaches used for deduplication over encrypted data are studied in this paper.

II. background

Deduplication

Deduplication is basically a compression technique for removing redundant data. Fig 1 explains the deduplication process before storing data onto memory. Deduplication can be categorized as file level deduplication and block level deduplication based on granularity. File level deduplication takes into account the entire file, thus even small update or append makes the file different from previous version of it and thereby reducing deduplication ratio. Where as in case of block level deduplication data blocks are considered for deduplication. Deduplication can further categorized based on location of deduplication i.e., as client side deduplication and as source side deduplication. Performing deduplication at client side is ensures bandwidth saving since only hash value of file is sent to server, if duplicate is existing [6] , [7].Deduplication is widely is used various applications like backup, metadata management, primary storage, etc. for storage optimization [8].

Convergent Encryption

Convergent encryption [2], is an encryption approach that support deduplication. With convergent encryption, encryption key is generated out of hash of plain text. Thus applying these technique identical plaintexts would produce same cipher text, and this helps in performing deduplication further.

Proof of Ownership

Deduplication works by computing cryptographic hash function on to data and using this hash value to determine similar data. Once a duplicate copy is found then new data is not uploaded but pointer to file ownership is updated thus saving storage and bandwidth. When it comes to client side deduplication, hash values of data are computed at client and send for duplicate check. An attacker, who gains access to hash value of a data which not authorized to him/her, may claim deduplication of file and thereby gaining access to the file. To defend such an attack, a Proof Of Ownership (PoW) has been proposed in [10], and various works like [10],[11], etc adapted this method. PoW works as an interactive algorithm between two parties - a prover and verifier to prove the ownership of the file. Verifier computes a short value of data M whereas, a prover need to compute short value of M and send it to verifier for claiming ownership of M [9],[10].

III. Related Works

Bellare et.al [12] propose an encryption scheme wherein key for encryption and decryption are derived from message itself.MLE key generation algorithm maps the message M to a key K and further the encryption algorithm generates cipher text C of the message using key K. Cipher text C is then mapped to a tag T, and this tag used for duplicate check by server. Keys used in MLE scheme are of fixed and shorter length thus does not result in much storage overhead.Chen et.al [13] put forward a method to achieve dual level source based deduplication of large encrypted files with block key management and Proof of Ownership [10],[11]. Author claims that MLE scheme were proposed for target based file level deduplication and extending it to dual level deduplication requires much metadata management. In BL-MLE scheme with the given input file , a master key is generated and set of block keys for each message block in the file .With tag generation algorithms file tags and block tags are generated and further these tags are used checking equality of blocks and files ensuring security to it. Ownership of files or blocks proved and verified by using PowPrf and PowVrf algorithms in this approach.

In [14] encryption and decryption data is performed at client side and key for this is provided by key server located at cloud storage provider premises. Homomorphic encryption is used as the one of key management scheme in this approach. Data encryption key is first computed by the initial file up loader and further distributed consequent verified up loader by key server. Data encryption key used for encryption are further encrypted with the hash of file content. Data encrypted with data encryption keys are send to the storage server. HEDup ensures privacy while enabling deduplication. Key server discussed in this approach may become a bottleneck when number of clients increase in case of large scale deployment, and a decentralized deployment of key server is supposed as a solution.

In [15] Bellare et. al claim that Message locked encryption [13] are subject to Brute force attack and proposes a new architecture called DupLess where Brute force is resisted. Client receives message based keys, for encryption, from key server via a Oblivious Pseudorandom function (OPRF) protocol. With OPRF public key for encryption is shared among clients where as secret key resides with key server. With this method attackers cost of attack increased and chance is eliminated.

Puzio et.al in [16] propose ClouDedup, a secure and efficient storage service which assures block level deduplication[7] data confidentiality at the same time using convergent key encryption[2] added with block level key management .Architecture of ClouDedup proposes to prevent well known attacks against convergent encryption by embedding a user authentication mechanisms and access control mechanisms. Thus, a server encryption is applied on top of convergent encryption performed by user. For each data segment a signature is linked to it , and need to be verified for retrieving it. To deal with block level key management a metadata manager(MM) has been added to architecture.MM uses file table- to store meta data about file, pointer table-to manage storage and a signature table- to store meta data about signature for meta data management.

S. Bugiel et.al in [17] proposes an approach that mainly involves two components - a trusted cloud and a commodity cloud. Trusted cloud is responsible for encrypting data and verifying operations performed on the commodity cloud. Security critical operations are performed by trusted cloud and queries to outsourced data are processed by commodity cloud. This approach claims protection against various security issues like leakage of data, computation manipulations, etc.

In [11] Li et.al propose a hybrid cloud approach to ensure security in deduplication which involves private cloud for providing tokens to access encrypted data in cloud. Data encryption technique employed here is convergent encryption [2] and PoW [9], [10] is used to ensure ownership eligibility to deduplicate the file. In

[18] M. W. Storer et.al proposes to provide single server and distributed storage systems with data security and space savings. With this method key for encryption is generated out of data chunk. Even a full compromise of the system cannot reveal which data chunk is owned which user since the decryption information is encrypted with client's private keys. Two models for secure deduplication Authenticated model and Anonymous model are used in this method. An authenticated model is similar Convergent key construction [2]. Anonymous model hides identities of both authors and readers.

IV. Implementation

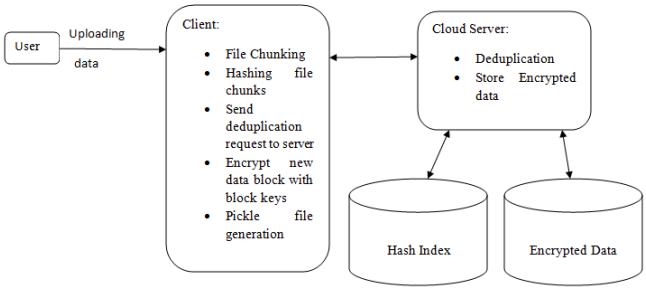

Figure 1: System Architecture

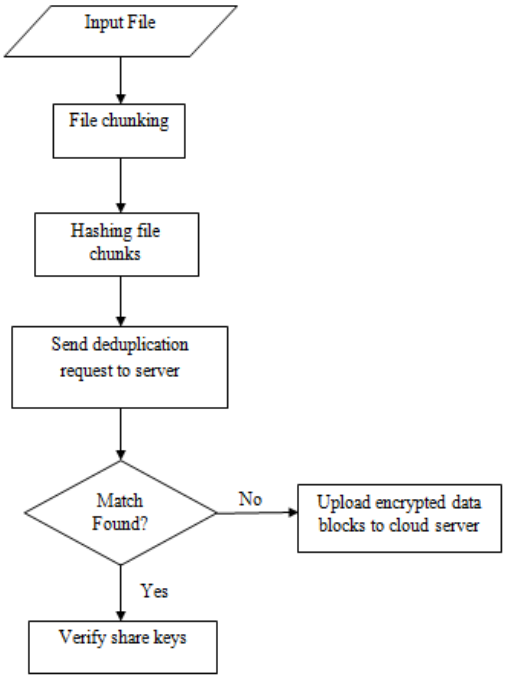

Figure 2: Client side operations

Two main components involved in are a client program and a storage server. Major functions that come under client program are file chunking followed by hashing and convergent encryption. The main function of cloud storage server is to store encrypted data as well as facilitating deduplication. Figure 1 shows the architecture of the system. Figure 2 shows the client side operation s supporting deduplication.

File chunking-Block generation

For each input file, the binary representation of file is divided into fixed sized blocks. Size of data block determines the level of granularity of deduplication. As the size of data block decreases level of deduplication increases meanwhile it may give rise to complex metadata management. On the other hand, larger sized blocks results in minimal Meta data management but lesser level of deduplication. Thus it is essential to find an optimal data block size for a fine balance between deduplication and Meta data management. In the experimental setup, we considered data block sized 4KB, 10KB, 25 KB over different set of data files of sizes 10MB, 50MB, 100MB and 200MB.And we observed that as the size of data block increases number of duplicate data identified and deduplication ratio decreases. Whereas, as time taken to perform deduplication

decreases with the increase in data block sizes. Thus, in order to obtain an optimal performance in terms of deduplication ratio as well as time, we chose 10 KB as our data block size.

Block key genertation and hashing

Data subject to hashing algorithm produces a unique data identifier which enables to identify the input from other pieces of data. Data hashing helps in deduplication
check to figure whether two data pieces are duplicates or not by comparing content hashes of aforesaid data pieces. Once content blocks are identified ,these blocks undergoes a strong hashing algorithm, in this paper, we use double SHA hashing algorithm to find unique hash of the input content blocks.

Convergent Encryption

Convergent encryption, also known as content hash keying, is a cryptosystem that produces identical cipher text from identical plaintext files. This has applications in cloud computing to remove duplicate files from storage without the provider having access to the encryption keys. Normally, when cloud services encrypt data, they use their own encryption key. With convergent encryption, the encryption key is derived from the file itself. As such, it produces identical cipher text from identical plaintext files. The system gained additional visibility in 2011 when cloud storage provider Bitcasa announced they were using convergent encryption to enable deduplication of data in their cloud storage service.

Pickle file generation

Pickle file is used for serializing and de-serializing a Python object structure. Any object in python can be pickled so that it can be saved on disk. Pickling is a way
to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script. Here for constructing pickle file out of block key and file content hash along with hash of encrypted data. The sole purpose of Pickle file generation is to restore the file by users. Pickle file generated is kept in premises of client machine.

Deduplication

Data deduplication involves finding and removing duplication within data without compromising its fidelity or integrity. The goal is to store more data in less space by segmenting files into small sized chunks, identifying duplicate chunks, and maintaining a single copy of each chunk. Redundant copies of the chunk are replaced by a reference to the single copy. In most organizations, the storage systems contain duplicate copies of many pieces of data. For example, the same file may be saved in several different places by different users, or two or more files that aren't identical may still include much of the same data. Deduplication eliminates these extra copies by saving just one copy of the data and replacing the other copies with pointers that lead back to the original copy. Data deduplication can be performed based two methods either by checking similarity signature of file or by checking similarity between content hashes, provided a strong content hashing technique is used so that no two different data contents produce same hash. Here we employ content hash similarity checking method, which produces more meaningful level of deduplication.

Backup and restore

The term Backup refers to copying or storing an additional copy of data facilitating disaster recovery from data loss or unintended manipulation of data. Encrypted content hash with block key is stored as backup file. Each file in the selected folder is processed to generate deduplicated backup files of the corresponding files in the folder. File path and an authorization password are two input that are to be provided for backing up data. For restoring file pickle file and password that is used for backup must be provided and backed up files will be getting reconstructed with the help of pickle file and content hashes.

V. Performance evaluation

Block level Deduplication

We conduct test bed evaluation of project. Our evaluation is based on number of duplicate copies bypassed on saving data to storage server. Our evaluation is performed on system equipped with Intel Core i3-5005U CPU 2.00 GHz with 4GB RAM and installed Ubuntu 14.04 ,64 -bit Operating System. The machine is connected to Cloud server hosted with domain name dedup.mtech-cse.in via Internet. The effectiveness of data deduplication is often expressed as a deduplication or reduction ratio, denoting the ratio of protected capacity to the actual physical capacity stored. A 10:1 ratio means that 10 times more data is protected than the physical space required to store it, and a 20:1 ratio means that 20 times more data can be protected. Factoring in data

growth, retention and assuming deduplication ratios in the 20:1 range, 2 TB of storage capacity could protect up to 40 TB of retained backup data [27].

To evaluate the proposed system we consider different file sets sized 10 MB,50 MB,100MBand 200 MB. And performed backing up multiple times. Corresponding space saving and time taken are observed. Deduplication ratio in percentage is obtained by calculating protected saving from observed value. Table 1 depicts the deduplication ratio obtained for different sized files using the proposed method with local server and Figure 3 shows corresponding Deduplication ratio graph. Table 2 shows the time taken for first time and second time backups and Figure 4 shows the time graph of first and second time backups.

Table 1: Evaluation of deduplication ratio obtained for different file sizes

File Sizes(MB)	Deduplication ratio(%)
10	2
50	11
100	40.8
200	50.8

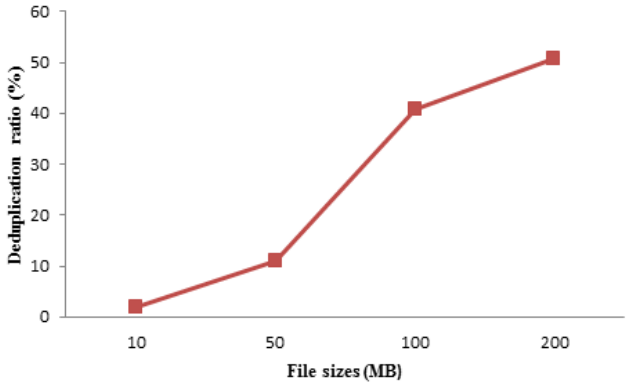

Figure 3: Deduplication ratio graph

Table 2: Time taken for first and second time backup process

File Sizes(MB)	Time taken in first time backing up(s)	Time taken in second time backing up(s)
10	187	8
50	960	39
100	1095	77
200	2117	121

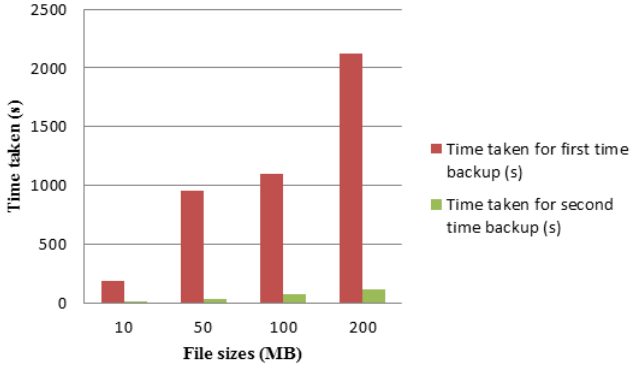

Figure 4: Time Graph

Security

In the proposed scheme Cloud storage server is considered to be semi trusted entity performing minimal set of operations. Additional encryption method employed by the server ensures that data is no longer vulnerable to limitations of convergent encryption such as dictionary attacks. In fact, additional key used for encryption prevents any other component in performing dictionary attacks on the data stored at the cloud storage service provider. And keying materials are assumed to safe kept at client for decrypting the data.

VI. Conclusion

Deduplication was introduced to cloud storage for saving bandwidth and storage capacity. Since most of the users rely on encrypting their data for protecting it from honest-but-curious cloud service providers. Encryption should be in such a way to enable deduplication. Various deduplication strategies over encrypted data studied in literature survey. Most of the deduplication strategies works on basis of convergent encryption, and convergent encryption makes deduplication compatible with encrypted data. This paper aims to make deduplication compatible with encrypted data by using convergent key encryption. As an better deduplication strategy, fixed block level deduplication is used in this paper. For ensuring security of deduplication in a multi user scenario, a block key and share key management is performed here. Thus, using fixed block level deduplication make sure that, in multi user cloud backup scenario, deduplication is performed well on encrypted data at the same time it ensures security by using block keys and share keys on top other security measures.

References

[1] Akhila, K., Amal Ganesh, and C. Sunitha. "A Study on Deduplication Techniques over Encrypted Data." *Procedia Computer Science* 87 2016 – Elsevier : 38-43.

[2] J. Douceur, A. Adya, W. Bolosky, D. Simon, and M. Theimer. "Reclaiming space from duplicate files in a serverless distributed file system. In Distributed Computing Systems", 2002. Proceedings. 22nd International Conference on, pages 617{624. IEEE, 2002.

[3] Dropbox http://www.dropbox.com.

[4] AmazonS3 http://aws/amazon.com/s3s.

[5] GoogleDrive http://www.drive.google.com.

[6] SNIA,"Advanced Deduplication Concepts,"[online] 2011.Available from http://www.snia.org/sites/default/education/tutorials/2011/fall/DataProtectionManagement/ThomasRiveria_Advanced_Dedupe_Concepts_FINAL.pdf

[7] http://searchdatabackup.techtarget.com/tip/Where-and-how-to-use-data-deduplication-technology-in-disk-based-backup

[8] Dutch T Meyer and William J Bolosky."A study of practical Deduplication". *ACM Transactions on Storage (TOS)*, 7(4):14, 2012.

[9] 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

[10] Yang, Chao, Jianfeng Ma, and Jian Ren. "Provable Ownership of Encrypted Files in De-Duplication Cloud Storage." *Ad Hoc & Sensor Wireless Networks*26.1-4 (2015): 43-72.

[11] Li, Jin, Yan Kit Li, Xiaofeng Chen, Patrick PC Lee, and Wenjing Lou. "A hybrid cloud approach for secure authorized deduplication." *Parallel and Distributed Systems, IEEE Transactions on* 26, no. 5 (2015): 1206-1216.

[12] Bellare, Mihir, Sriram Keelveedhi, and Thomas Ristenpart. "Message-locked encryption and secure deduplication." *Advances in Cryptology–EUROCRYPT 2013*. Springer Berlin Heidelberg, 2013. 296-312.

[13] Chen, Rongmao, Yi Mu, Guomin Yang, and Fuchun Guo. "BL-MLE: Block-Level Message-Locked Encryption for Secure Large File Deduplication." (2015). *Information Forensics and Security, IEEE Transactions on* 26(2015), no. 12: 2643-2652.

[14] Miguel, Rodel, and Khin Mi Mi Aung. "HEDup: Secure Deduplication with Homomorphic Encryption." In *Networking, Architecture and Storage (NAS), 2015 IEEE International Conference on*, pp. 215-223. IEEE, 2015.

[15] Bellare, Mihir, Sriram Keelveedhi, and Thomas Ristenpart. "Dupless: Server-aided encryption for deduplicated storage." *Proceedings of the 22nd USENIX conference on security*. USENIX Association, 2013.

[16] Puzio, Pasquale, Refik Molva, Melek Önen, and Sergio Loureiro."ClouDedup:Secure Deduplication with Encrypted Data for Cloud Storage." .In *Cloud Computing Technology and Science(CloudCom),2013 IEEE 5th International Conference on (Volume:1)p.363 – 370.*

[17] 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.

[18] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, "Secure data deduplication," in *Proc. 4th ACM Int. Workshop Storage Security Survivability*, 2008, pp. 1–10.

[19] Li, Jie, Xia Chen, Xumin Huang, Song Tang, Yingmeng Xiang, Mehdi Hassan, and Abdul Hameed Alelaiwi. "Secure Distributed Deduplication Systems with Improved Reliability."*Computers, IEEE Transactions on* 64, no. 12(2015): 3569 – 3579.

[20] G. R. Blakley and C. Meadows, "Security of ramp schemes," in *Proc. Adv. Cryptol*., 1985, vol. 196, pp. 242–268.

[21] A.D. Santis and B. Masucci, ''Multiple Ramp Schemes,'' *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1720-1728, July 1999.

[22] Zhou, Yukun, Dan Feng, Wen Xia, Min Fu, Fangting Huang, Yucheng Zhang, and Chunguang Li. "SecDep: A User-Aware Efficient Fine- Grained Secure Deduplication Scheme with Multi-Level Key Management." *Mass Storage Systems and Technologies (MSST), 2015 31st Symposium on*, pp. 1-14.

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

[24] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage," *Tech. Rep. IBM Research, Zurich*, ZUR 1308-022, 2013.

[25] 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.

[26] Li, Jin, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick PC Lee, and Wenjing Lou. "Secure deduplication with efficient and reliable convergent key management." *Parallel and Distributed Systems, IEEE Transactions on* 25, no. 6 (2014): 1615-1625.

[27] http://searchdatabackup.techtarget.com