International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 09 | Sep 2020 www.irjet.net p-ISSN: 2395-0072

Secure Deduplication using Dekey

Kalyani R. Gawande[1], Prof. V. B. Bhagat[2]

[1]M.E.Student, Dept. of Computer Science and Engineering, P.R.Pote College of Engg. Amravati, MH, India
[2]Professor, Computer Science and Engineering, P.R.Pote College of Engg. Amravati, MH, India

---

Abstract - *Data deduplication is a technique for reducing duplicate copies of information in cloud storage. By using Deduplication, process moment and storage space capability in cloud are decreased in efficient approach. Since it is, an occurring challenge is to perform protected de duplication in cloud storage. Plaintexts are encrypted to create convergent key and cipher text. The human or computer without using the convergent key can't read by the cipher text. Convergent key encryption is used to encrypt analogous information copies with similar cipher text and identical convergent key. We first establish a baseline approach in which every consumer grips a self-governing master key used for encrypting the convergent keys and outsourcing them to the cloud. Though, such a baseline key organization method creates a huge numeral of keys along with the growing total number of consumers and requires consumers toward contributed secure the master keys. Dekey is a new construction in which consumer do not necessitate to handle all keys on their individual except as an alternative securely allocate the convergent key distributes transversely various servers. A new-fangled enhanced Ramp secret sharing scheme and proof of Ownership (PoW) is used for dekey encryption and decryption.*

Key Words: Cloud Computing Storage Space Deduplication Convergent Encryption Dekey

1. INTRODUCTION

Cloud Storage is a service where data is fully handled, managed, and backed up. It allows the user to store files so that the user can access them from any location. The provider company makes them available to the users online by keeping the upload the files on an external server. This will gave the companies using cloud storage services ease and convenience, but it can be costly. Users should also be aware that backing up their data is still required when using cloud storage services, because recovering data from cloud is much slower than local backup.

The encrypted convergent keys are then store up, along with the corresponding encrypted data copies, in cloud storage space. The master key can be used to recuperate the encrypted keys and thus the encrypted files. In this approach each consumer merely requests to maintain the master key and the metadata regarding the outsourced data.

Though, the baseline approach endures two dangerous exploitation problems. First, it is inefficient, as it will create a huge number of keys with the growing total number of consumers. Specially, every consumer should correlate an encrypted convergent key with every block of its outsourced encrypted data copies; hence as to later on reinstate the data copies. While different consumers may distribute the similar data copies, they should have their individual set of convergent keys so that other users cannot access their files. Since a result, the number of convergent keys being initiated linearly balances with the blocks being stored and the number of consumers.

2. RELATED WORK

2.1 Traditional encryption

To secure the confidentiality of outsourced information, a variety of cryptographic resolutions have been proposed in the literature. Their schemes construct on traditional (symmetric) encryption, in which every consumer encrypts information with an independent secret key. A few studies propose to make use of threshold secret distribution to sustain the forcefulness of key management. Though, the above studies do not regard as deduplication. By means of traditional encryption, various consumers will basically encrypt the same data copies with their personal keys, but this will lead to various ciphertexts and therefore create deduplicatio.

2.2 Proposed Work

We propose for providing security in both insider and outsider attacker and monitoring them we use for that Dekey, user behaviour profiling and Decoy Technology. Dekey is a new method in which users do not need to control any keys on their own but instead securely distribute the convergent key shares across multiple servers. Dekey using the Ramp secret sharing scheme and indicate that Dekey incurs limited overhead in realistic environments we propose a new construction called Dekey, which provides efficiency and reliability guarantees for convergent key management on both user and cloud storage sides. To provide efficient and reliable convergent key management through convergent key Deduplication and secret sharing. Dekey bear both file-level and block level Deduplication. Security analysis demonstrates that Dekey is reliable in terms of the definitions specified in the proposed security model. In particular, Dekey remains secure even the contestant controls a limited number of key servers. We implement Dekey using the Ramp secret sharing scheme that permit the key management to adapt to different reliability

and confidentiality levels. Our evaluation demonstrates that Dekey incurs ristricted overhead in normal upload/download operations in realistic cloud environments.

3. RAMP SECRETE SHARING

Dekey make use of the Ramp secret sharing scheme (RSSS) to store convergent keys. Specially, the RSSS creates n distributes from a secret such the secret can be improved from any k distributes but cannot be improved from less than k distributes, and no data about the secret can be presumed from any r distributes. It is identified that when r = 0, the (n,K,0)-RSSS grow to be the (n,k) Rabin's Information Dispersal Algorithm (IDA) when r =k-1, the (n,k,k-1)-RSSS grow to be the (n,k) Shamir's Secret Sharing Scheme (SSSS). The (n,k,r)-RSSS makes on two primordial functions:

1. Share splits a secret S into elements of equal size, creates r arbitrary elements of the equal size, and encodes the k elements using a non-systematic k-ofn erasure code1 into n distributes of the equal size;
2. Recover obtains any k out of n distributes as inputs and then outputs the inventive secret S.

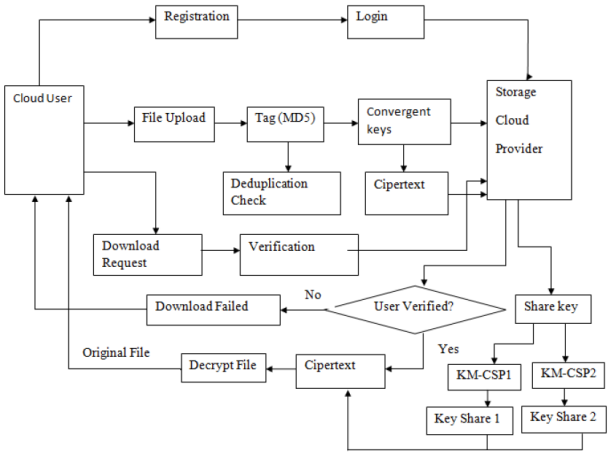


Fig-1: Architecture of Dekey

4. SECURE DEDUPLICATION

Subject to chunking/blocking method, two types of deduplication method/strategies are there:

4.1. File level chunking: This algorithm, examines whole file as one chunk and does not split the file into small blocks hence in this method, for the whole file only one index value is generated and this index value, is further compared with previously saved index values. As there is only one index value for every file, there would be relatively lesser entries in the index table. This concept would decrease the total storage size and in the same index table more entries for unique indexes can be made as well. This file level segregation fails when there is a slight change in file data, because it will generate index for complete file again rather, it should generate index for only changed data which in turns decreases the ratio of deduplication elimination and throughput of the system.

4.2 Block Level Chunking: It is of two types as given below:

4.2.1 Fixed-Size Chunking: Using this technique the data file is further partitioned into fixed sized blocks or chunks. Fixed size chunking solves the problem which had arisen in file level chunking as in this method the index value is generated for different blocks instead of files. Therefore when any data is changed index of only that block is changed not of the whole. On other hand many small chunks are created for large files resulting in consumption of more storage space for large number of index value and metadata.

4.2.2 Variable-Size Chunking: Using this technique, the data file is partitioned in numerous small sized blocks or chunks which are rather of variables size than being of same fixed size and the file is segregated on the basis of the content of the data than same fixed size value. Therefore solving the problem and eliminating the drawbacks of the fixed size chunking. It is to be noted that in fixed sized chunking data boundaries do not change even when there is a change in the data whereas in later i.e. variable sized chunking data boundaries are of variable size depending upon the different parameters and even these boundaries can be shifted when there is any change in file some deletion of data or file occurs. Therefore when there is any change in fie, fewer boundaries are need to be change.

5. RESULT ANALYSIS

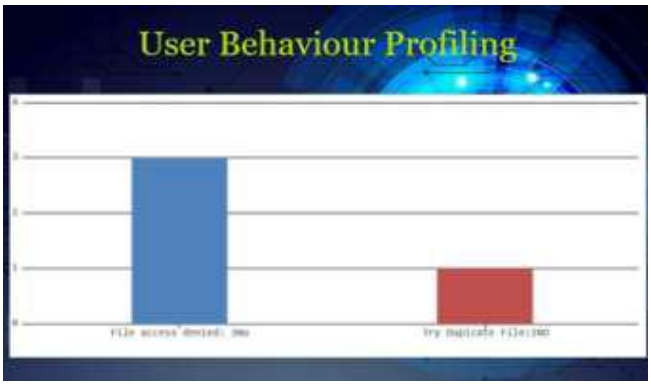
We analyzed our datasets around multiple aspects for dedup space savings and used those findings to design our primary data deduplication system. In this section, we evaluate some other aspects of our data deduplication system that are related to post-processing deduplication. Post-processing deduplication throughput. Using the dataset, we examined post-processing deduplication throughput, calculated as the amount of original data processed per second. An entry-level HP ProLiant SE326M1 system with one quad-core Intel Xeon 2.27 GHz L5520 and 4 GB of RAM was used, with a 3- way RAID-0 dynamic volume on top of three 1TB SATA 7200 RPM drives. To perform an windows-to-windows comparison, we ran a post-processing deduplication session multiple times with different indexing options in a controlled environment. Each deduplication session uses a single thread. Moreover, we ensured that there were no CPU-intensive tasks running in parallel for increased measurement accuracy. The baseline case uses a regular index where the full hash (SHA-256, 32 bytes) and location information (16 bytes) for each unique chunk is stored in

RAM. The optimized index uses the RAM space efficient design.

Second approach for Data Security we used JFreeChart tool for the detection of masquerade activity in graphical format of user profiling behavior and decoy

Table-1: User Profiling Detection

Table-2: Graphical User profiling detection



6. CONCLUSION

We can limit the damage of stolen information if we decrease the value of that stolen information to the attacker. We can achieving this through a „preventive" disinformation attack. We posit that secure deduplication services can be implemented given additional security features insider attacker on Deduplication and outsider attacker by using the detection of masquerade activity. By the combination of these security features will provide unprecedented levels of security for the deduplication.

7. FUTURE SCOPE

Identity management system: Cloud computing users are identified and used their identities for accessing the services. A secure trust based identity management scheme is essentially a need by all cloud service provider and users.

Secure trust based Solution for cloud computing Service: A secure environment for execution of the cloud computing services along with overall security considerations is a challenge. A secure and trusted solution is the requirement that needs.

REFERENCES

[1] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation. Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997.

[2] Ayushi "A Symmetric Key Cryptographic Algorithm "International Journal of Computer Applications (0975 - 8887) ©2010 Volume 1 – No. 15

[3] Abdul Wahid Soomro, Nizamuddin, ArifIqbal Umar, Noorul Amin." Secured Symmetric Key Cryptographic Algorithm for Small Amount of Data" 3rd International Conference on Computer & Emerging Technologies (ICCET 2013)

[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 Proc. ICDCS, 2002, pp. 617-624.

[5] W. J. Bolosky, J. R. Douceur, D. Ely, and M. Theimer, "Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs", SIGMETRICS 2000, ACM, 2000, pp.34-43.

[6] A. Adya, W. J. Bolosky, M. Castro, R. Chaiken, G. Cermak, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, and R. Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI), Boston, MA, Dec.2002.USENIX.