Article can be accessed online at http://www.publishingindia.com

Reliable Deduplication of Encrypted Data in Cloud Computing

Santhosh R*, Kalaiyarasi K**

Abstract

Cloud computing is a concept which is popular among not only software professionals but also common internet users. It allows a program to be executed on multiple connected machines at the same time over a network. Deduplication is one way of ensuring that the network and storage overhead is minimized. Digital data is growing exponentially and by removing redundancy deduplication technique achieves this. There are many deduplication schemes proposed but these schemes only focuses on the files without encryption. Convergent encryption allows the cloud to enable deduplication on encrypted files. However, it is vulnerable to dictionary attacks. In this paper, we propose a new scheme to address this issue. Instead of deriving the encryption key from the entire content, encryption key will be derived from each block of the content. Encrypted password of the user will be appended with the file content to make the file unique among the users. This will allow the files to be protected from the confirmation of the file attack.

Keywords: Attacks, Cloud, Deduplication, Encryption, Storage

1. Introduction

Cloud computing is a model which provides services over internet. There are three main categories which classifies these services as platform as a service, software as a service and infrastructure as a service. Organization which outsources the resources such as server, hardware and storage etc., provides infrastructure as service[9]. Cloud computing is based on utility computing, virtualization and distributed computing. It is also based on the networking and software services. Cloud uses virtualization techniques to implement resource partitioning. Cloud users deploy virtual machines (VMs) on data center resources to accomplish lesser computation time [2]. Digital data is growing exponentially and by removing redundancy, deduplication technique achieves minimal network and storage overhead. Deduplication eliminates the same copy of content by storing only one physical copy and referring other redundant data to that physical copy, instead of storing multiple copies of the same data [1]. Commonly cloud storage offerings are among multiple users and deduplication techniques are most effective in this case. It reduces bandwidth and space requirements of cloud storage services [7]. Deduplication can be categorized as file-level deduplication and block-level deduplication in terms of size. File-level deduplication detects redundancy between the files uploaded and removes the redundant copy from the cloud storage. Block-level deduplication detects and removes redundancy between different data blocks. The content can be divided into smaller data blocks. It may be fixed-size or variable-size blocks. Fixed-size block reduces the computation overhead(on block boundaries). Better deduplication efficiency can be obtained by using variable-size blocks [1]. Convergent encryption [3] ensures data privacy in deduplication. But this will allow the content prone to dictionary attacks and confirmation of file attack. To overcome this, we propose a new scheme in which the encryption key will be derived

* Assistant Professor, Department of Computer Science and Engineering, Faculty of Engineering, Karpagam University, India. Email: santhoshrd@gmail.com

** PG Scholar, Department of Computer Science and Engineering, Faculty of Engineering, Karpagam University, India. E-mail: kalaiyarasi.karuppusamy@gmail.com

from each block of the content instead of entire content. When the file is uploaded into cloud, encrypted password of the user will be appended to the content. This will make the content unique for the user. Then, the content along with the encrypted password will be split into blocks and each block will be encrypted. This will allow the files to be protected from the "Confirmation of the file" attack.

2. Related Works

SecCloud and SecCloud+ are two secure systems proposed in [4] which aim to achieve data integrity and deduplication in cloud. Auditing entity, part of SecCloud helps clients generate data tags before uploading as well as audit the integrity of data having been stored in cloud. The previous issue of computational load is too large for tag generation has been fixed by this design. On both block level and sector level, auditing functionality designed in SecCloud is supported. Using a proof of ownership protocol between clients and cloud servers, SecCloud enables secure deduplication. Customers always want their data to be encrypted before uploading. SecCloud+ is designed keeping this in mind. A secret "seed" controls the convergent key of file during Convergent encryption. This ensures that the convergent key cannot be directly derived from the file content. Thus the dictionary attack is prevented [4].

In two-party computations, using simulation-based framework, the security of private data deduplication protocols is framed. The standard cryptographic assumptions are presented/analyzed based on construction of private deduplication protocols in [5].

A scheme based on content popularity is detailed in [6]. The encryption scheme guarantees security for unpopular data. It also provides better bandwidth and storage benefits for popular data. This ensures, data deduplication is effective for popular data, while unpopular data is secured by encryption. A song or video requires less protection when compared to a private document. The threshold value could be removed when unique copies of an unpopular file have been uploaded. When this takes place, (i) the security for the popular file is lowered from semantic to standard convergent, and (ii) the convergent encryption layer remains as it is. Thus the security is compromised for storage efficiency. When the file becomes popular, space is restored for the uploaded copies. This allows deduplication to take place normally for further uploads[6].

A threshold based scheme is discussed in [7]. The correlation between the files in the cloud and deduplication is weakened to reduce the risks in deduplication. A threshold value will be assigned randomly for every file and then deduplication will be applied to the file only when the number of copy is more than the assigned threshold value [7].

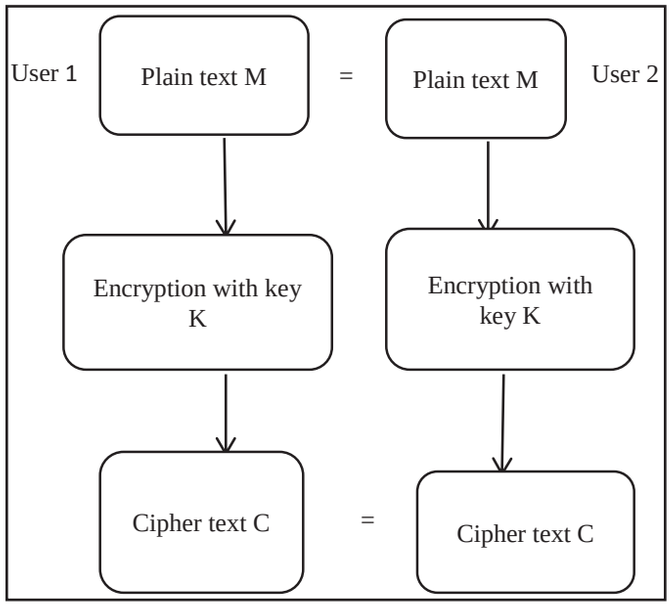
Dekey method is a convergent key management which is considered as reliable and efficient in secure deduplication for managing the convergent key in cloud storage [10].

ClouDedup is an efficient and secure storage service which has been proposed in [8]. It provides both data confidentiality and block-level deduplication at the same time. It makes the system more efficient and flexible. Since this provides block-level deduplication, it requires a new component for implementing key management along with deduplication. The additional overhead of this key management is minimal and the computational and storage cost is not impacted [8].

3. Proposed System

Convergent Encryption (CE) derives the encryption key from plaintext. The simple implementation of convergent encryption is depicted in Fig 1.

Figure 1. Convergent Encryption



By applying this technique, two users with two identical plain texts will obtain two identical cipher texts since

the encryption key is the same; hence the cloud storage provider will be able to perform deduplication on such cipher texts.

The proposed work is to increase the reliability of the deduplication performed on encrypted files. Using convergent encryption allows the cloud to enable deduplication. However, it is vulnerable to dictionary attacks. Instead of deriving the encryption key from the entire file content, encryption key will be derived from each block of the content. When the file is uploaded into cloud, the content will be appended with the encrypted password of the user. This will be done using the encryption key generated by the user. Then, the content along with the encrypted password will be split into blocks and each block will be encrypted. This will allow the files to be protected from the "Confirmation of the file" attack. It also means that only block level deduplication will be possible even if two users upload the same file.

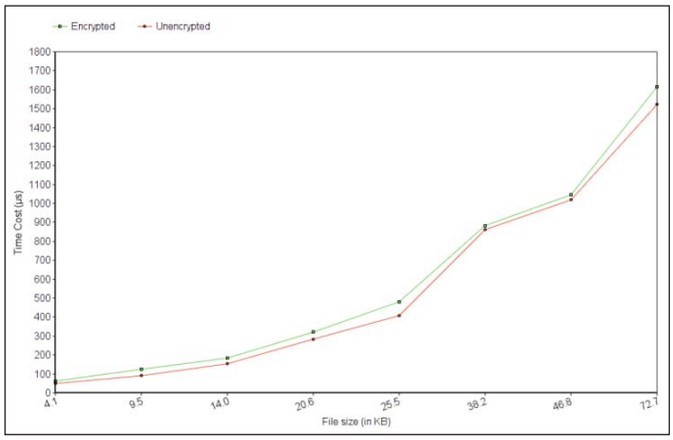
The proposed scheme appends encrypted password at the end of the file before splitting it into fixed size blocks. Then, each block is encrypted with different encryption key. Deduplication is applied for the encrypted content once it is uploaded into cloud storage.

4. Experimental Results

The experiment is performed in Intel core i3 system with 2.9 GHZ processor and on Windows 7 OS using CloudSim 3.0.3 simulator. CloudSim is a library which helps to simulate the cloud environment in a single system. It allows the user to create data centers, hosts, virtual machines, storage and other components virtually. This helps to test the concept before the actual implementation. Using the class VM, two virtual machine components are created. The property values are 2GB RAM, MIPS as 1000 in each VM. The experiment is performed for files which are of varying size like 4162, 9534, 14013, 20624, 25496, 38210, 46802 and 72739 (in bytes) respectively. Deduplication for plain and cipher texts have been implemented for comparison.

From the fig 2, we can observe that the time cost of encrypted content and plain content have a similar pattern. The proposed system does not add any performance overhead while providing secure deduplication in efficient way.

Figure 2. Impact of Time Cost on Encrypted/ Unencrypted Content



5. Conclusion and Future Work

We propose in this paper a scheme which allows cloud storage to overcome the dictionary and confirmation of file attack while applying deduplication on encrypted data. It preserves data confidentiality. This scheme is simple and easy to implement. But this limits to block level deduplication. File level deduplication is not possible with this scheme. The following issue will be addressed in our future works: First, we will extend our deduplication scheme to multiple hosts. Second, we will enhance the model to variable-size content block deduplication. Third, we will try to improve the performance of the proposed model further.

References

1. Li, J., Chen, X., Huang, X., Tang, S., & Xiang, Y., Hassan, M. M., & Alelaiwi, A. (2015). *Secure Distributed Deduplication Systems with Improved Reliability*. IEEE Transactions on Computers.
2. Nicolae, B., & Cappello, F. (2013). Blobcr: Virtual disk based check point restart for HPC applications on IAAS clouds. *Journal of Parallel and Distributed Computing*, 73(5), 698-711.
3. Douceur, J. R., Adya, A., Bolosky, W. J., Simon, D., & Theimer, M. (2002). *Reclaiming space from duplicate files in a serverless distributed file system*. Proceedings on 22nd International Conference on Distributed Computing Systems, (pp. 617-624).
4. Li, J., Li, J., Xie, D., & Cai, Z. (2015). *Secure Auditing and De-duplicating Data in Cloud*. IEEE Transactions on Computers.

5. Ng, W. K., Wen, Y., & Zhu, H. (2012). *Private data de-duplication protocols in cloud storage*. Proceedings of the 27th Annual ACM Symposium on Applied Computing. New York, NY, USA.
6. Stanek, J., Sorniottiy, A., Androulakiy, E., & Kencl, L. (2006). A secure data de-duplication scheme for cloud storage. *Parallel Computing*, 32(5), 331-356.
7. Harnik, D., Pinkas, B., & Shulman-Peleg, A. (2011). *Side channels in cloud services, the case of deduplication in cloud storage*. IEEE Transactions on Parallel and Distributed Systems, 9(3), 272-284.
8. Puzio, P., Molva, R., Onen, M., & Loureiro, S. (2013). *ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage*. IEEE Transactions on Computers, 62(5), 990-1003.
9. Zheng, Q., & Xu, S. (2012). *Secure and efficient proof of storage with de-duplication*. In Proceedings of the second ACM Conference on Data and Application Security and Privacy, ser. CODASPY'12. New York, NY, USA: ACM, (pp. 1-12).
10. Kakariya, G., & Rangdale, S. (2014). A hybrid cloud approach for secure authorized de-duplication. *International Journal of Computer Engineering and Applications*, 8(1), 2576-2579.