IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

An Intelligent Compression: A Reliable Space Optimization in Cloud Storage

Shyma Manzoor[1], Amrutha Chandran[2], Raji Jayan[3], Nayana P[4], Archana R S[5], Sowmya K S[6], Jooby.E[7]

UG Scholar, Department of CSE, College of Engineering, Perumon, Kollam, Kerala, India[1,2,3,4,5]

Assistant professor, Department of CSE and IT, College of Engineering, Perumon, Kollam, Kerala, India[6]

Assistant Professor, Department of CSE, College of Engineering, Perumon, Kollam, Kerala India[7]

Abstract: The deduplication of data is the technique used for avoiding the redundancy of data by placing only one physical copy in the cloud storage. The data uploaded by the users are compared to the history of data in the cloud storage and it is ideal for highly redundant operations like backup which requires repeatedly copying and storing the same data set a number of times for purpose of recovery. To protect the data and to ensure confidentiality while supporting deduplication, convergent encryption technique has been designed to encrypt the data and this encrypted data will be stored in the cloud. The concept of convergent key is used to ensure the confidentiality and authenticity of the secure deduplication system. The concept of master key is used which enables us to protect the convergent key sent by the third party to the users and this ensures the authenticity of the convergent key Dekey is the derived key, which is the second approach, in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across the hash table.

Keywords: Confidentiality, Authenticity, Convergent encryption, Key management, Deduplication.

I. INTRODUCTION

Cloud computing is an emerging computing technology, also known as 'on-demand computing', where shared resources, data and information are provided to computers and also for other devices on-demand. Cloud computing has now become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance, scalability, accessibility as well as availability. As cloud computing provides storage space, a large amount of redundant data is being stored and shared by users with specified privilege, which define the access rights of the stored data. one significant challenge of cloud storage services is the management of the ever-increasing volume of data.

Cloud computing provides a low cost, scalable, location independent infrastructure for data management and storage. The rapid adoption of cloud services is accompanied by increasing volumes of data stored at remote servers. Hence, techniques for saving disk space and network bandwidth are needed.

A central upcoming concept in this context is deduplication, where the server stores a single copy of each file, in spite of how many users store that file. All users that store the file merely use pointer to the single copy of the file stored at the cloud. Moreover, if the server already has a copy of the file then clients do not even need to store it again to the server, thus saving bandwidth as well as storage.

In a typical secure deduplication system, the user first uploads the data, the third party auditor, which is a trusted one, computes the hash value from the data which is uploaded by the user and this hash value acts as the convergent key and the third party auditor checks if that hash value already exists in the hash table. If the hash value is not in the hash table, then the entire file will be stored in the cloud in an encrypted form and the corresponding convergent key will be stored in the hash table. Otherwise, since the file already exists ,the convergent key will be in the hash table (potentially uploaded by someone else),then the convergent key generated for the already uploaded file will be shared to the current user thereby only one physical copy exists in the cloud. That is, only one physical copy of that redundant file is stored in the cloud storage

II.ABOUT THE SYSTEM

The Farsite distributed file system was introduced by J.R. Douceur, A. Adya, W.J. Bolosky, D. Simon, and M. Theimer in 2002 which provides availability by replicating each file onto multiple desktop computers. In the view of the fact that this replication consumes considerable storage space, it is essential to reclaim used space where possible. Measurement of over 500 desktop file systems shows that nearly half of all consumed space is occupied by duplicate files. In these concept, identification and distribution identical files in the Farsite distributed file system, for the purpose of reclaiming storage space consumed by incidentally redundant content. Farsite is a secure, scalable, serverless file system that logically functions as a centralized file server. In this Farsite concept, four steps are performed. The First step is: Enabling the identification and distributing of identical files when these files are (for security reasons) encrypted,the second step is:Identifying, in a scalable, fault tolerant manner, files that have identical content,the third step is relocating the replicas of files with identical content to a common set of

storage machines,i.e,Farsite and the final step is distributing the identical files to reclaim storage space, while maintaining the semantics of separate files.The disadvantage of Farsite concept is: provide additional expense, physical plant, administration, and vulnerability to geographically localized faults. So there is need to present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication.[4]

In 2010, P. Anderson and L. Zhang described an algorithm which takes advantage of the data which is common between users to increase the speed of backups, and reduce the storage requirements. This algorithm supports client-end per-user encryption which is necessary for confidential personal data. It also supports a unique feature which allows immediate detection of common subtree, avoiding the need to query the backup system for every files.The disadvantage of the existing system lies in the case of Backups. That is, here, Backups are often made to a local disk and copies are not stored offsite. Backups are not encrypted and vulnerable to theft. [2]

D. Harnik, B. Pinkas, and A. Shulman-Peleg, in 2010 demonstrated how de-duplication can be used as a side channel which reveals information about the contents of files of other users. In a different scenario, de-duplication can be used as a covert channel by which malicious software can communicate with its control center, regardless of any firewall settings at the attacked machine. Due to the high savings offered by cross-user deduplication, cloud storage providers are unlikely to stop using this technology. So they propose simple mechanisms that enable cross-user de-duplication while greatly reducing the risk of data leakage.The disadvantage in this existing system is that the block level and file level deduplication is only performed. File level Deduplicaton is also known as single-instance storage. In the file level deduplication, File-level data deduplication compares a file that has to be archived or backup that has already been stored by checking all its attributes against the index. The File level data deduplication is also difficult in the case of large files as it more difficult to compare all the attributes of a large file. In, Block-level de-duplication, data deduplication operates on the basis of sub-file level. As the name implies, that the file is being broken into segments blocks or chunks that will be examined for previously stored information for redundancy. The block level deduplication becomes difficult in the case of large files, that is, when the number of blocks increases. [6]

Later in 2012, M. Bellare, S. Keelveedhi, and T. Ristenpart proposed Message-Locked Encryption (MLE), where the key under which encryption and decryption are performed is itself derived from the message. MLE provides a way to achieve secure de-duplication, a goal currently targeted by numerous cloud-storage providers. MLE is a primitive of both practical and theoretical concern. Achieving MLE here is easy because we can use part of the message as the key to encrypt the other part. The problem with traditional encryption is that exchanging of secret keys them over the Internet or a large network while preventing them from falling into the wrong hands but anyone who knows the secret key can decrypt the message. This system does not consider data privacy. [3]

In 2014, Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick P.C. Lee, and Wenjing Lou proposed Dekey. Dekey applies deduplication among convergent keys and distributes convergent key shares across multiple key servers, while preserving semantic security of convergent keys and confidentiality of outsourced data. [1]

III. PROPOSED SYSTEM

In this, the concept of Convergent key is used, where, while realizing deduplication it also provides a viable option to enforce data confidentiality. The data copy is encrypted or decrypted with the convergent key, which is derived. This is done by computing the cryptographic hash value of the content of the data copy itself.When the key generation process is completed, users gets the convergent keys and the encrypted data will be stored in the cloud. As the encryption performed is deterministic, identical data copies will generate the same convergent key which allows the secure deduplication system to perform deduplication in an efficient way. The updation operation is not yet discussed in cloud storage which is also now introduced.

The Update operation in the Secure Deduplication process is another contribution which is very challenging one as it provides the users a facility to update the file they have uploaded .So, there is a chance for arising 4 cases:

Case1:
When the user wants to update a file then the user must verify whether the file to be updated is shared or not. If it is shared then the file which is to be updated cannot be deleted from the cloud as the convergent key is shared to some other users and the user should upload the updation as a new file.

Case2:
When the user wants to update a file then it must be verified whether the file to be updated is shared or not. If it is not shared then the file which is to be updated is deleted from the cloud as the convergent key is not shared to some other users and the user should upload the updation as a new file by deleting the existing file from the cloud.

Case3:
When the user wants to update a file then it must verify whether the file to be updated is shared or not. If it is not shared then the file which is to be updated is deleted from the cloud as the convergent key is not shared to some other users and the user can upload the updation with the same name of the existing file. If it is shared then the file which is to be updated cannot be deleted from the cloud as the convergent key is shared to some other users and the user should upload the updation as a new file. Then after the user uploaded the updated file then it must also verify whether the file already exists in the cloud storage .If it already exists then the convergent key will be shared to this user and this file cannot be deleted by any users for

DOI 10.17148/IJARCCE.2016.5269

further updation and also the cloud filename of the existing file will be also be shared to this user. So that the filename will be same as the uploaded file name and only the cloud filename is changed.

Case4:

When the user wants to update a file then the user must verify whether the file to be updated is shared or not. If it is not shared then the file which is to be updated can be deleted from the cloud as the convergent key is not shared to some other users and the user can upload the updations with the same name of the existing file. If it is shared, then the file which is to be updated cannot be deleted from the cloud as the convergent key is shared to some other users and the user should upload the updation as a new file. Then after the user uploaded the updated file the user must also verify whether the file already exists in the cloud storage .If the file not already exists in the cloud storage, then the convergent key will be generated for this user by the third party auditor will be stored in the hash table and also the file will be stored in the cloud after encryption. Also, this file can be deleted by any users for further updation and can be uploaded as a new file.

Advantages: The data uploaded can only be decrypted by the corresponding data owners with their convergent key which is inturn decrypted by using the master key.The secure deduplication system is more efficient and secured.The backups are also stored in encrypted form.

The main entities in the proposed system are namely:

- User
- Trusted Party Auditor
- Administrator

User: User can upload and download files. The main advantage of the proposed system is the provision to update the uploaded files.

Third Party Auditor: Third Party Auditor is a trusted party registered by the administrator to the secure deduplication system. The TPA can view the user uploads, user requests and the hash table. The main functionality of the TPA is that they view the data uploaded by the user and generates the convergent key using secure hash algorithm, that is, a hash value which is stored in the hash table. The TPA uses this convergent key for determining the duplicate files in the secure deduplication system. If the uploaded file is a duplicate file, then the TPA shares the convergent key to the User else TPA generate the convergent key for the new file uploaded and forwards the convergent key to the administrator.

Administrator: manages the data uploaded by the user, administrator controls and approves the third party auditor. The Administrator provides master key to the user. The administrator can view the TPA requests, cloud contents and the hash table.

IV.CONCLUSION

The main aim of secure data deduplication is to protect the security of the data by ensuring different copies of the file in the cloud storage by performing duplicate check and eliminating the redundant copies. The Third Party Auditor is allowed to perform the duplicate check for files having same convergent key. Deduplication can be performed on files uploaded by the user to the secure deduplication system. Deduplication can be done at these files uploaded by the user, which reduces the upload bandwidth on the network. By using deduplication technique, storage space can be saved by eliminating duplicate copies in the cloud storage system.The convergent key generation on user files helps for effectively implementing deduplication technique. Dekey applies deduplication among convergent keys and distributes convergent key shares across hash table, while preserving semantic security of convergent keys and confidentiality of outsourced data. Dekey Implementation that incurs small encoding/decoding overhead in the regular upload/download operations.

REFERENCES

[1] "Secure De-duplication with Efficient and Reliable Convergent Key Management" Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick P.C. Lee, and Wenjing Lou in IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 6, June 2014

[2] P. Anderson and L. Zhang, ''Fast and Secure Laptop Backups with Encrypted De-Duplication,'' in Proc. USENIX LISA, 2010, pp. 1-8.

[3] M. Bellare, S. Keelveedhi, and T. Ristenpart, ''Message-Locked Encryption and Secure Deduplication,'' in Proc. IACR Cryptology ePrint Archive, 2012, pp. 296-3122012:631.

[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] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman- Peleg, ''Proofs of Ownership in Remote Storage Systems,'' in Proc. ACM Conf. Comput. Commun. Security, Y. Chen, G. Danezis, and V. Shmatikov, Eds., 2011, pp. 491-500.

[6] D. Harnik, B. Pinkas, and A. Shulman-Peleg, ''Side Channels in Cloud Services: De-duplication in Cloud Storage,'' IEEE Security Privacy, vol. 8, no. 6, pp. 40-47, Nov. /Dec. 2010.