Asian Journal of Applied Science and Technology (AJAST)
(Open Access Quarterly International Journal) Volume 2, Issue 2, Pages 840-845, April-June 2018

Heterogeneous Data Storage Management with Deduplication in Cloud Computing

Sangeetha A[1] and Geetha K[2]

[1,2]*Department of CSE, Excel Engineering College, Tamilnadu, India.*

Article Received: 01 March 2018 *Article Accepted: 09 April 2018* *Article Published: 28 April 2018*

ABSTRACT

Cloud storage as one of the most important services of cloud computing helps cloud users break the bottleneck of restricted resources and expand their storage without upgrading their devices. In order to guarantee the security and privacy of cloud users, data are always outsourced in an encrypted form. 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. However, encrypted data could incur much waste of cloud storage and complicate data sharing among authorized users. We are still facing challenges on encrypted data storage and management with deduplication. Traditional deduplication schemes always focus on specific application scenarios, in which the deduplication is completely controlled by either data owners or cloud servers. In this project, propose a heterogeneous data storage management scheme, which flexibly offers both deduplication management and access control at the same time across multiple Cloud Service Providers. In this project, the original data copy is first encrypted with a convergent key derived by the data copy itself, and the convergent key is then encrypted by a master key that will be kept locally and securely by each user. The encrypted convergent keys are then stored, along with the corresponding encrypted data copies, in cloud storage. In addition, the project also considers the revocation of users in the given group. If the original (first) user of the group intimates the server with a user's (B) revocation, then the server rejects the proof of ownership submitted by that user (B). Likewise, session based deduplication is considered. The project is developed using Microsoft Visual Studio .Net 2005 as front end. The coding language used is Visual C# .Net. MS-SQL Server 2000 is used as back end database.

Keywords: Data deduplication, Cloud computing, Storage management, Access control.

1. INTRODUCTION

Cloud computing means that instead of all the computer hardware and software you're using sitting on your desktop, or somewhere inside your company's network, it's provided for you as a service by another company and accessed over the Internet, usually in a completely seamless way. Exactly where the hardware and software is located and how it all works doesn't matter to you, the user—it's just in the nebulous "cloud" that the Internet represents. Cloud computing is a buzzword that means different things to different people. For some, it's just another way of describing IT (information technology) "outsourcing"; others use it to mean any computing service provided over the Internet or a similar network; and some define it as any bought-in computer service you use that sits outside your firewall. However we define cloud computing, there's no doubt it makes most sense when we stop talking about abstract definitions and look at some simple, real examples—so let's do just that.

2. RELATED WORK

One critical challenge of today's cloud storage services is the management of the ever increasing volume of data. To make data management scalable, deduplication has been a well-known technique to reduce storage space and upload bandwidth in cloud storage. Instead of keeping multiple data copies with the same content, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Each such copy can be defined based on different granularities: it may refer to either a whole file (i.e., file level deduplication), or a more fine-grained fixed-size or variable-size data block (i.e., block-level deduplication). Today's commercial cloud storage services, such as Dropbox, Mozy, and Memopal, have been applying deduplication to user data to save maintenance cost. According to the user's view, data outsourcing raises security and privacy concerns.

We must trust third-party cloud providers to properly enforce confidentiality, integrity checking, and access control mechanisms against any insider and outsider attacks. However, deduplication, while improving storage and bandwidth efficiency, is incompatible with traditional encryption. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different ciphertexts, making deduplication impossible. A new construction Dekey is proposed to provide efficient and reliable convergent key management through convergent key deduplication and secret sharing. Dekey supports both file-level and block level de-duplications.

2.1. METHODOLOGY

The resulting intensive key management overhead of the existing system leads to the huge storage cost, as users must be billed for storing the large number of keys in the cloud under the pay-as-you-go model. So that, to avoid the huge storage cost, a new system will be proposed. If the master key is accidentally lost, then the user data cannot be recovered; if it is compromised by attackers, then the user data will be leaked. To achieving secure de-duplication in efficiency and reliability guarantees for convergent key management on both user and cloud storage sides, the new system scheme will be proposed. Using the proposed algorithms, the KM-CSP can efficiently manage the group of users. In addition, the project also considers the revocation of users in the given group. If the original (first) user of the group intimates the server with a user's (B) revocation, then the server rejects the proof of ownership submitted by that user (B). Likewise, session based deduplication is considered. Here if the user provides the session duration i.e, front date and to date, then only with the data range, proof of ownership can be allowed in server on those dates. This increases the security if the outsourced data need to be safely accessed on the given duration.

2.2. ANTICIPATED SYSTEM MODEL

This study makes new construction Dekey to supply economical and reliable focused key management through focused key deduplication and secret sharing. Dekey supports each file-level and block-level deduplications. Security analysis is demonstrates that Dekey is secure in terms of the definitions laid out in the planned security model. In specific, Dekey remains secure even the person controls a restricted variety of key servers. They implement Dekey exploitation the Ramp secret sharing theme that allows the key management to adapt to completely different responsibleness and confidentiality levels. Their analysis demonstrates that Dekey incurs restricted overhead in traditional upload/download operations in realistic cloud environments. Symmetric cryptography uses a standard secret key to encipher and decode info. Since the key used for this project square measure terribly weak, the prevailing system is a smaller amount secure. User revocation management isn't enforced. The key will be management solely among the cluster members.

3. ALGORITHM

KeyGenCE (M): K is the key generation algorithm that maps a data copy M to a convergent key K;EncryptCE(K, M): C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs

and then outputs a ciphertext C;DecryptCE(K,C): M is the decryption algorithm that takes both the ciphertext C and the convergent key K as inputs and then outputs the original data copy M TagGenCE(M): T(M) is the tag generation algorithm that maps the original data copy M and outputs a tag T(M). We allow TagGenCE to generate a tag from the corresponding ciphertext, by using T(M)=TagGenCE(C), where C=EncryptCE(K,M).

4. PROBLEM DEFINITION

This project makes new construction Dekey to provide efficient and reliable convergent key management. In particular, Dekey remains secure even the adversary controls a limited number of key servers. They implement Dekey using the Ramp secret sharing scheme that enables the key management to adapt to different reliability and confidentiality levels.

Their evaluation demonstrates that Dekey incurs limited overhead in normal upload/download operations in realistic cloud environments. Symmetric encryption uses a common secret key to encrypt and decrypt information. Since the key used for this project are very weak, the existing system is less secure. User revocation management is not implemented. The key can be management only within the group members. To overcome the drawbacks of this existing system this project is designed using Java as front end and SQL server 2005 as back end.

S.No	Number Communication User	Number of Social Workers	Average of attack Finding Time in Existing system (ms)	Average of attack Finding Time in Proposed system (ms)
1	5	25	33	29
2	10	30	35	31
3	15	35	42	39
4	20	40	54	48
5	25	45	63	57
6	30	50	72	68
7	35	55	81	75

The following Fig 1.2 describes experimental result for comparison between existing and proposed system for in social network using average time taken finding attacker discovery. The figure contains average attacker finding, number of user working details, number of social worker and average of attacker occur finding in existing system and average of attacker finding in proposed system details are shown.

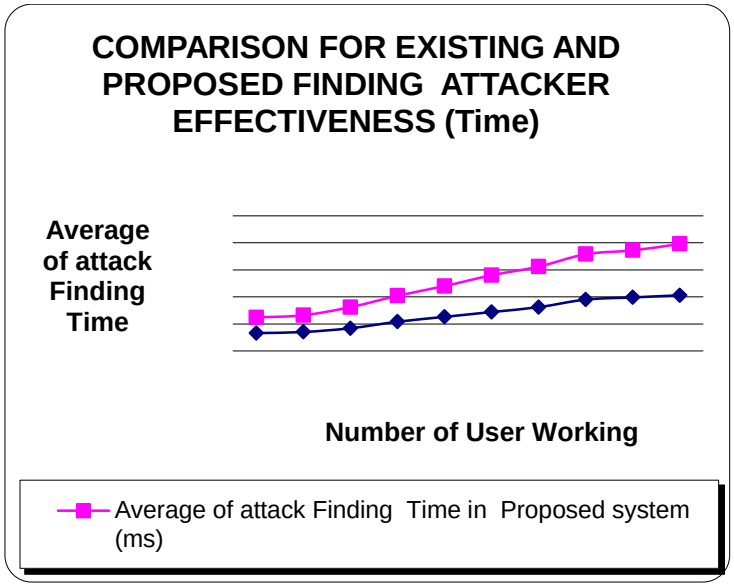

Fig 1.2 Comparison for existing and proposed system in Finding Attacker effectiveness (Time)

5. SYMMETRIC ENCRYPTION

In this work, a symmetric encryption/decryption technique is devised. Symmetric encryption uses a common secret key K to encrypt and decrypt information. A symmetric encryption scheme consists of following functions:

1. KeyGenSE (1λ):): K is the key generation algorithm that generates K using security parameter 1・;
2. EncryptSE (K, M): C is the symmetric encryption algorithm that takes the secret K and message M and then outputs the ciphertext C;
3. DecryptSE (K, C): M is the symmetric decryption algorithm that takes the secret K and ciphertext C and then outputs the original message M.

5.1. Proposed Framework

The resulting intensive key management overhead of the existing system leads to the huge storage cost, as users must be billed for storing the large number of keys in the cloud under the pay-as-you-go model. So that, to avoid the huge storage cost, a new system will be proposed. If the master key is accidentally lost, then the user data cannot be recovered; if it is compromised by attackers, then the user data will be leaked. To achieving secure de-duplication in efficiency and reliability guarantees for convergent key management on both user and cloud storage sides, the new system scheme will be proposed

1. KeyGenCE (M): K is the key generation algorithm that maps a data copy M to a convergent key K;
2. EncryptCE(K, M): C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a ciphertext C;
3. DecryptCE(K,C): M is the decryption algorithm that takes both the ciphertext C and the convergent key K as inputs and then outputs the original data copy M.

4. TagGenCE(M): T(M) is the tag generation algorithm that maps the original data copy M and outputs a tag T(M). We allow TagGenCE to generate a tag from the corresponding ciphertext, by using T(M)=TagGenCE(C), where C=EncryptCE(K,M).

5.1.1. CONVERGENT ENCRYPTION

Localized encryption gives information classification in deduplication. A client (or information proprietor) gets a united key from every unique information duplicate and scrambles the information duplicate with the concurrent key. Moreover, the client determines a tag for the information duplicate, with the end goal that the tag will be utilized to recognize copies. To distinguish copies, the client initially sends the tag to the server side to check if the indistinguishable duplicate has been as of now put away. Note that both the joined key and the tag are autonomously inferred and the tag can't be utilized to derive the united key and bargain information secrecy. Both the scrambled information duplicate and its relating tag will be put away on the server side.

6. DEDUPLICATION PROTOCOL

Assumes that a server typically has to handle a huge number of files and the files themselves are stored on a secondary storage with a large access time. The server can store only a small amount of data per file in fast storage but it cannot aff ord to retrieve the file or parts of it from secondary storage upon every upload request. As a result, the private data deduplication scheme must allow the server to store only an extremely short information per file that will enable it to check claims from clients that they have that file without having to fetch the file contents for verification.

New notion which they call private data deduplication protocols is introduced and formalized in the context of two-party computations. A feasible result of private data deduplication protocols has been proposed and analyzed. They have shown that the proposed private data deduplication protocol is provably secure in the simulation-based framework assuming that the underlying hash function is collision-resilient, the discrete logarithm is hard and the erasure coding algorithm can erasure up to α-fraction of the bits in the presence of malicious adversaries.

7. CONCLUSION

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. This project attempts to formally address the problem of achieving efficient and reliable key management in secure deduplication. It introduces a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud.

In addition, the users can revoked from the given group at any time. Likewise, session based deduplication is considered so that it increases the security for the outsourced data. The secure deduplication with heterogeneous data storage management method provides flexible cloud data storage without duplication and better access

control. To concurrently handle multiple audit sessions from different users for their outsourced data files, further it is extend into multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency.

REFERENCES

[1] Q. Liu, C. C. Tan, J. Wu, and G. Wang, "Efficient information retrieval for ranked queries in cost-effective cloud environments," in Proc. 2012 IEEE INFOCOM, pp. 2581-2585, 2012.

[2] M. Zhou, Y. Mu, W. Susilo, M. H. Au, and J. Yan, "Privacy preserved access control for cloud computing," in Proc. of IEEE 10th Int. Conf. Trust, Secur. Privacy Comput. Commun., pp. 83-90, 2011.

[3] M. Bellare, S. Keelveedhi, and T. Ristenpart, "DupLESS: server aided encryption for deduplicated storage," in Proc. of 22nd USENIX Conf. Secur., pp. 179-194, 2013.

[4] T.-Y. Wu, J.-S. Pan, and C.-F. Lin, "Improving accessing efficiency of cloud storage using de-duplication and feedback schemes," IEEE Systems J., vol. 8, no. 1, pp. 208-218, 2014.

[5] Z. Yan, W. X. Ding, and H. Q. Zhu, "A scheme to manage encrypted data storage with deduplication in cloud," in Proc. Of ICA3PP2015, pp. 547-561: Springer, 2015.

[6] Z. Yan, W. X. Ding, X. X. Yu, H. Q. Zhu, and R. H. Deng, "Deduplication on encrypted big data in cloud," IEEE Trans. On Big Data, vol. 2, no. 2, pp. 138-150, April-June 2016.

[7] Z. Yan, M. J. Wang, Y. X. Li, and A. V. Vasilakos, "Encrypted data management with deduplication in cloud computing," IEEE Cloud Comput. Mag., vol. 3, no. 2, pp. 28-35, 2016.

[8] J. Hur; D. Koo; Y. Shin; and K. Kang, "Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage," IEEE Trans. Knowl. Data Eng., vol. 28, no. 11, pp.3113-3125, 2016.

[9] J. Li, Y. K. Li, X. F. Chen, P. P. C. Lee, and W. J. Lou. "A hybrid cloud approach for secure authorized deduplication," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1206-1216, 2015.

[10] J. Liu, N. Asokan, and B. Pinkas. "Secure deduplication of encrypted data without additional independent servers," in Proc. of 22nd ACM SIGSAC Conf. Comput. Commun. Secur., pp. 874-885. ACM, 2015.