www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 11 November, 2014 Page No. 9201-9204

A Literature Survey On Secure De-Duplication Using Convergent Encryption Key Management

Ms. Madhuri A. Kavade[1], Prof. A.C.Lomte[2]

[1] Computer Department
JSPM's BSIOTR,Pune
madhuri.kavade@gmail.com

[2] Computer Department
JSPM's BSIOTR,Pune
archanalomte@gmail.com

ABSTRACT: ***One vital challenge of today's cloud storage services is the management of the ever-increasing quantity of data. To make data management scalable, de-duplication has been a well-known technique to condense storage space and upload bandwidth in cloud storage. Instead of keeping multiple data copies with the same content, de-duplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy.***

Now a day the most arising challenge is to perform secure de-duplication in cloud storage. Although convergent encryption has been extensively adopted for secure de-duplication, a critical issue of making convergent encryption practical is to efficiently and reliably manage a huge number of convergent keys. We first introduce a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master keys which is inefficient an unreliable. For that purpose we are going to formally address the problem of achieving efficient and reliable key management in secure de-duplication. We propose Dekey, a new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers.

Keywords: De-duplication, Convergent Keys

1. Introduction

Cloud computing is getting more and more popular as it can provide low-cost and on demand use of vast storage and processing resources. With the explosive growth of online digital contents, cloud storage focuses on effectively coalescing storage resources for better power utilization and cost effectiveness. As the volume of data grows, also increasing is the Total Cost of Ownership (TCO), which includes storage infrastructure cost, management cost and human administration cost. Therefore in cloud storage systems, reducing the amount of data that need to be transferred, stored, and managed becomes a crucial, and it also benefits for application performance, storage costs and administrative overheads. As a result, Data De-duplication is an important and popular cost-saving feature for cloud storage. The term data de-duplication refers to techniques that store only a single copy of redundant data, and provide links to that copy instead of storing other actual copies of this data. With the transition of services from tape to disk, data de-duplication has become a key component in the backup process. By storing and transmitting only a single copy of duplicate data, de-duplication offers savings of both disk space and network bandwidth.

In today's cloud storage services one of the significant challenges is the management of the ever-increasing amount of data. According to the analysis report of IDC, the amount of data is expected to reach 40 trillion gigabytes in 2020 [5]. With the continuous increase of the number of users and the size of their data, data de-duplication becomes more and more a necessity for cloud storage providers. The simple idea behind de-duplication is to store duplicate data (either files or blocks) only once. Therefore, if a user wants to upload a file (block) which is already stored, the cloud provider will add the user to the owner list of that file (block). De-duplication has proved to accomplish high space and cost savings and many cloud storage providers are currently adopting it. De-duplication is a illustrious technique to reduce storage space and upload bandwidth and has been used to make data management scalable.

As an alternative of keeping numerous data copies with the identical content, de-duplication eliminates surplus data by keeping only one physical copy and referring other surplus data to that copy. There are two types of de-duplication one is file-level de-duplication and

another is block-level de-duplication. Among that file-level de-duplication refers to the whole file whereas block-level de-duplication refers to the fixed or variable size data block.

To make de-duplication secure we have to apply certain security mechanism like encryption. Traditional encryption requires different users to encrypt their data with their own keys, so identical data copies of different users will lead to different ciphertext and for this reason de-duplication is incompatible with traditional encryption.

Convergent encryption [4] provides a possible option to implement data confidentiality while realizing de-duplication. Convergent encryption, a cryptosystem that produces indistinguishable ciphertext files from the same plaintext files, irrespective of their encryption keys.It encrypts/decrypts a data with a convergent key, which is derived by computing the cryptographic hash value of the content of the data copy itself [4]. After key generation and data encryption, users retain the keys and send the ciphertext to the cloud. Since encryption is deterministic, identical data copies will generate the same convergent key and the same ciphertext. This allows the cloud to perform de-duplication on the ciphertexts. The ciphertexts can only be decrypted by the corresponding data owners with their convergent keys.

We have two approaches baseline approach and Dekey approach. By using baseline approach we can understand how convergent encryption realizes de-duplication. 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. The master key can be used to recover the encrypted keys and hence the encrypted files. In this way, each user only needs to keep the master key and the metadata about the outsourced data.

There are two problems with baseline approach. First, it is inefficient, because it generates enormous number of keys with the increasing number of users. In particular, each user must correlate an encrypted convergent key with each block of its outsourced encrypted data copies, so as to later on re-establish the data copies. Although different users may share the same data copies, they must have their own set of convergent keys so that no other users can access their files. As a result, the number of convergent keys being introduced linearly balance with the number of blocks being stored and the number of users.[1]

Second, it is unreliable, it requires each user to dedicatedly protect his own master key and if master key is accidently lost, then user data can't be recovered. To avoid these problems we propose Dekey approach where efficient and reliable key management is the main motivation behind proposing Dekey approach.

2. Motivation

- With the potentially infinite storage space offered by cloud providers, users tend to use as much space as they can.
- The vendors constantly look for techniques aimed to minimize redundant data and maximize space savings.
- So the technique which can give both of these features must have to implement. This motivates us to introduce a technique called data de-duplication.
- The simple idea behind de-duplication is to store duplicate data (either files or blocks) only once. Therefore, if a user wants to upload a file (block) which is already stored, the cloud provider will add the user to the owner list of that file (block).
- De-duplication has proved to achieve high space and cost savings and many cloud storage providers are currently adopting it.
- On the other hand, de-duplication introduces new security risks. So there is need of secure de-duplication.

3. Literature Survey

Literature review is the process of presenting the summary of the journal articles, conference papers and study resources. So in this section we have studied the related topics an summarized it below.

According to the data granularity, de-duplication strategies can be categorized into two main categories: file-level de-duplication and block-level de-duplication , which is nowadays the most common strategy. In block-based de-duplication, the block size can either be fixed or variable. Another categorization criteria is the location at which de-duplication is performed if data are de-duplicated at the client, then it is called source-based de-duplication, otherwise target-based.

In source-based de-duplication, the client first hashes each data segment he wishes to upload and sends these results to the storage provider to check whether such data are already stored: thus only "not de-duplicated" data segments will be actually uploaded by the user. While de-duplication at the client side can achieve bandwidth savings, it unfortunately can make the system vulnerable to side-channel attacks whereby attackers can immediately discover whether a certain data is stored or not. On the other hand, by de-duplicating data at the storage provider, the system is protected against side-channel attacks but such solution does not decrease the communication overhead.

Many people now store huge amount of personal and corporate data on laptops or home computers. These often have poor connectivity, and are susceptible to theft or hardware failure. Conventional backup solutions are not well suited to this environment. So client-end per-user encryption is necessary for confidential personal data. [2]

The Farsite distributed file system 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. So there is need to present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Our mechanism includes convergent encryption, which enables duplicate files to combine into the space of a single file, even if the files are encrypted with different users.[4]

Cloud storage services commonly use de-duplication, which eliminates redundant data by storing only a single copy of each file or block. De-duplication reduces the space and bandwidth requirements of data storage services, and is most effective when applied across multiple users, a common practice by cloud storage offerings. In this context they have 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 de-duplication,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. [7]

Throughout the past few years, a enormous number of online file storage services have been introduced. At the same time as several of these services provide basic functionality such as uploading and retrieving files by a specific user, more advanced services offer features such as shared folders, real-time association, and minimization of data transfers or unrestricted storage space. Overviews of existing file storage services and examine Dropbox, an advanced file storage solution, in depth. Based on the results they show that Dropbox is used to store copyright-protected files from a popular file sharing network [8]

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. [3]

Cloud storage systems are becoming increasingly popular. A technology that keeps their cost down is de-duplication, which stores only a single copy of redundant data. Client-side de-duplication attempts to recognize de-duplication opportunities already at the client and save the bandwidth of uploading copies of existing files to the server. Attacks that exploit client-side de-duplication, allowing an attacker to gain access to arbitrary-size files of other users based on a very small hash signatures of these files. More specifically, an attacker who knows the hash signature of a file can convince the storage service that it owns that file, hence the server lets the attacker download the entire file. [6]

4. Primitives

In this section, we formally define the cryptographic primitives used in our secure de-duplication.

4.1 Symmetric Encryption

Symmetric encryption uses a common secret key *k* to encrypt and decrypt information. A symmetric encryption scheme consists of three primitive functions:

- **KeyGen→ *k*** is the key generation algorithm that generates *k* using security parameter.
- **Encrypt(*k*,M) → C** is the symmetric encryption algorithm that takes the secret *k* and message M and then outputs the ciphertext C.
- **Decrypt(*k*,C)→ M** is the symmetric decryption algorithm that takes the secret and ciphertext C and then outputs the original message M.

4.2 Convergent Encryption

Convergent encryption provides data confidentiality in de-duplication. A user (or data owner) derives a convergent key from each original data copy and encrypts the data copy with the convergent key.

The basic idea of convergent encryption (CE) is to derive the encryption key from the hash of the plaintext. The simplest implementation of convergent encryption can be defined as follows:

Alice derives the encryption key from her message M such that K = H(M), where H is a cryptographic hash function; she can encrypt the message with this key, hence:

C = E(K;M) = E(H(M);M), where E is a block cipher.

By applying this technique, two users with two identical plaintexts will obtain two identical ciphertexts since the encryption key is the same; hence the cloud storage provider will be able to perform de-duplication on such ciphertexts. Furthermore, encryption keys are generated, retained and protected by users. As the encryption key is deterministically generated from the plaintext, users do not have to interact with each other for establishing an agreement on the key to encrypt a given plaintext. Therefore, convergent encryption seems to be a good candidate for the adoption of encryption and de-duplication in the cloud storage domain. In addition, the user derives a tag for the data copy, such that the tag will be used to detect duplicates. A convergent encryption scheme can be defined with four primitive functions:

- ✓ **KeyGen(M) → K** is the key generation algorithm that maps a data copy M to a convergent key K
- ✓ **Encrypt(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 cipher text C
- **Decrypt(K,C)→** M is the decryption algorithm that takes both the cipher text C and the convergent key K as inputs and then outputs the original data copy M
- **TagGen(M) →T(M)** is the tag generation algorithm that maps the original data copyM and outputs a tag

T(M). We allow TagGen to generate a tag from the corresponding cipher text by using T(M)=TagGen(C), where C=Encrypt(K,M).

5. System Model

The architecture includes three entities viz. the user, the storage cloud service provider (S-CSP), and the key management cloud service provider (KM-CSP). The task of each is as given below:

- **User:** A user is an individual that wants to outsource data storage to the S-CSP and access the data later. The user is allowed to upload unique data. And he/she don't have any right to upload any duplicate data, which may be owned by the same or different users.
- **S-CSP:** The S-CSP provides the data outsourcing service and stores data on behalf of the users. The S-CSP eliminates the storage of redundant data via de-duplication and keeps only unique data to reduce the storage cost.

- **KM-CSP:** A KM-CSP maintains convergent keys for users. Each convergent key is distributed across multiple KM-CSP to provide additional security. It provides service to handle the key management.

6. Conclusion

Baseline key management scheme generates an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master keys. So we propose Dekey, an efficient and reliable convergent key management scheme for secure de-duplication. Dekey applies de-duplication among convergent keys and distributes convergent key shares across multiple key servers, while preserving semantic security of convergent keys and confidentiality of outsourced data.

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 De-duplication,'' 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] J. Gantz and D. Reinsel, The Digital Universe in 2020: Big Data,Bigger Digital Shadows, Biggest Growth in the Far East, Dec.2012[Online]Available: *http://www.emc.com/collateral/analystreports/idc-the-digital-universe-in-2020.pdf*

[6] 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*.

[7] 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*.

[8] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl, ''Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space,'' in Proc. *USENIX Security, 2011, p. 5*.