ISSN 2395-1621

A Novel cloud approach for enhancing Security and De-Duplication

[#1]Ms. Gharat Trupti, [#2]Mr. Kamble Manojkumar, [#3]Ms. Shinde Prerana, [#4]Mr. Sawant Pradip

[1]twgharat@gmail.com
[2]manojkumarkamble90@gmail.com
[3]preranashinde42@gmail.com
[4]pradipsawant013@gmail.com

[#1234]Department of Computer engineering, UCOER, Pune.

ABSTRACT

As more corporate and private users outsource their data to cloud storage providers, recent data breach incidents make end-to end encryption an increasingly prominent requirement.Unfortunately, semantically secure encryption schemes render various cost-effective storage optimization techniques, such as data de-duplication, ineffective. We present a novel idea that differentiates data according to their popularity. Based on this idea, we design an encryption scheme that guarantees semantic security for unpopular data and provides weaker security and better storage and bandwidth benefits for popular data. This way, data de-duplication can be effective for popular data, whilst semantically secure encryption protects unpopular content.

ARTICLE INFO

Article History

Received :4th April 2016

Received in revised form :

5th April 2016

Accepted : 8th May 2016

Published online :

11th May 2016

Keywords—Deduplication, authorized duplicate check, confidentiality, hybrid cloud

I. INTRODUCTION

The basic idea of this project comes from the fact that cloud is big platform to store and to retrieve the data in huge capacity. Where there is greater possibility of duplication of the data can be happen due to this there will be huge storage space is used unnecessarily.So to avoid this problem proposed system put forwards an idea of avoiding this duplications based on maintaining the hash tags of the _les before encryption and conducting advance searches for the existed filles using some powerful concepts like bloom filter and inverted index to speed up the complete process.

Motivation of the Project

As more corporate and private users outsource their data to cloud storage providers, re-cent data breach incidents make end-to end encryption an increasingly prominent requirement.Unfortunately, semantically secure encryption schemes render various cost-effective storage optimization techniques, such as data de-duplication, ineffective. We present a novel idea that differentiates data according to their popularity. Based on this idea, we design an encryption scheme that guarantees semantic security for unpopular data and provides weaker security and better storage and bandwidth benefits for popular data. This way, data de-duplication can be effective for popular data, whilst semantically secure encryption protects unpopular content.

II. LITERATURE SURVEY

Unfortunately, de-duplication loses its effectiveness in conjunction with end to-end encryption. End-to-end encryption in a storage system is the process by which data is encrypted at its source prior to ingress into the storage system. It is becoming an increasingly prominent requirement due to both the number of security incidents linked to leakage of unencrypted data [1] and the tightening of sector-specific laws and regulations. Clearly, if semantically secure encryption is used, file de-duplication is impossible, as no one—apart from the owner of the decryption key—can decide whether two ciphertexts correspond to the same plaintext. Trivial solutions, such as forcing users to share encryption keys or using deterministic encryption, fall short of providing acceptable levels of security. Several de-duplication schemes have been

proposed by the research community [2–4] showing how de-duplication allows very appealing reductions in the usage of storage resources [5, 6]. Most works do not consider security as a concern for de-duplicating systems; recently however, Harnik et al. [7] have presented a number of attacks that can lead to data leakage in storage systems in which client-side de-duplication is in place. To thwart such attacks, the concept of proof of ownership has been introduced [8, 9]. None of these works, however, can provide real end-user confi- dentiality in presence of a malicious or honest-but-curious cloud provider. Convergent encryption is a cryptographic primitive introduced by Douceur et al. [10, 11], attempting to combine data confidentiality with the possibility of data de-duplication. Convergent encryption of a message consists of encrypting the plaintext using a deterministic (symmetric) encryption scheme with a key which is deterministically derived solely from the plaintext. Clearly, when two users independently attempt to encrypt the same file, they will generate the same ciphertext which can be easily deduplicated. Unfortunately, convergent encryption does not provide semantic security as it is vulnerable to content-guessing attacks. Later, Bellare et al. [12] formalized convergent encryption under the name message-locked encryption. As expected, the security analysis presented in [12] highlights that message-locked encryption offers confidentiality for unpredictable messages only, clearly failing to achieve semantic security. Xu et al. [13] present a PoW scheme allowing client-side de-duplication in a bounded leakage setting. They provide a security proof in a random oracle model for their solution, but do not address the problem of low min-entropy files. Recently, Bellare et al. presented DupLESS [14], a server-aided encryption for de-duplicated storage. Similarly to ours, their solution uses a modified convergent encryption scheme with the aid of a secure component for key generation. While DupLESS offers the possibility to securely use server-side de-duplication, our scheme targets secure client-side de-duplication.

I. Preliminaries

(A) Set Theory

- Let S={ } be as system for De-duplication
- Identify Input as D={ d_1, d_2, d_3,……d_n}
  Where d_i= no of Documents
  S= {D}
- Identify A_n as Output i.e. Annotation
  S= {D, A_n}
- Identify Process P
  S= {Q, P, A_n}
  P= {B_F, C_R, I_I, S_V, R_{CC}}
  Where B_F = Bloom filter
  C_R = Correlation
  I_I = Inverted index
  S_V = Subset vector
  R_{CC} = Reverse circle cipher

S = {D, B_F, C_R, I_I, S_V, R_{CC}, A_n}

(B) SET DESCRIPTION:

1. Bloom filter:

Set B_F:

B_{F0} = String file data

B_{F1} = MD5 hashing

B_{F02}= liner hash search

B_{F3} = identifying sequences

2. Correlation:

Set C_R:

$C_{R\,0}$ = Matching correlation vector

$C_{R\,1}$ = Covariance identification

$C_{R\,2}$ = correlation coefficient

3. Inverted index:

Set I_I:

I_{I0} = Data label tag

I_{I1} = File sequence index

I_{I2} = Index subset

I_{I3} = merging subset

4. Subset vector

Set S_V:

S_{V0} = File content hash key

S_{V1} = Matching user set

S_{V2} = Respective file name vector

S_{V3} = subset vector creation

5. Reverse circle cipher:

Set R_{cc}:

R_{cc0} = Read the data in string

R_{cc1} = Divide string into blocks

R_{cc2} =Consider block index

R_{cc3} =Calculate key factor in integer

R_{cc4} =get rotation factor

R_{cc5} =Rotate block according to rotation factor

R_{cc6} =Replace with special character

R_{cc7} =Concatenate block String

III. EXISTING SYSTEM

To make data management scalable in cloud computing, de-duplication has been a well-known technique and has attracted more and more attention recently. Data de-duplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. 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. De-duplication can take place at either the file level or the block level. For file level de-duplication, it eliminates duplicate copies of the same file. De-duplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files.

IV. PROPOSED SYSTEM

Convergent encryption has been proposed to enforce data confidentiality while making de-duplication feasible. It encrypts/decrypts a data copy with a *convergent key*, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the ciphertext to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies will generate same convergent key and hence the same ciphertext. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys. Thus, convergent encryption allows the cloud to perform de-duplication on the ciphertexts and the proof of ownership prevents the unauthorized user to access the file. However, previous de-duplication systems cannot support differential authorization duplicate check, which is important in many applications. In such an authorized de-duplication system, each user is issued a set of privileges during system initialization each file uploaded to cloud is also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files.

V. SYSTEM MODEL

System Architecture

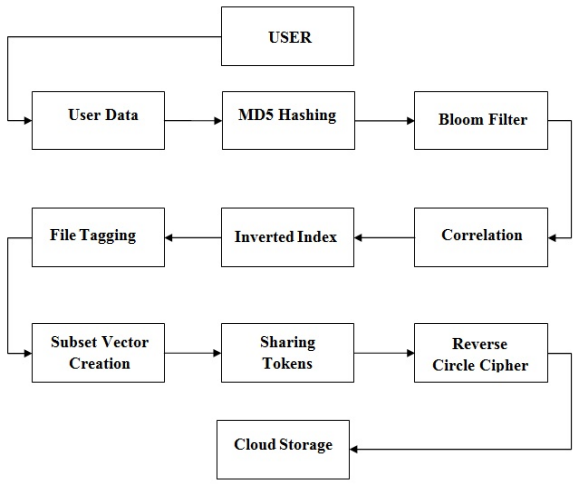

Fig: System Architecture

In this system architecture , describes the overall working of our system model. In this first on accepting user data MD5 hashing processing on given user's data . with the help of this MD5 hashing digest given data or message in 128-bit,In which same contents containing two or many files generates single hash key value for that. Then for the purpose of searching files in given data we proposed a bloom filtering. Then by using correlation is done for which files are related to each other or not? File tagging also helps to make easier searching. Inverted index is a search engine algorithm which helps for full text searches depending upon its location on database file. By using File vector creation creating smaller data set from larger data. Tokens are used for securing the data purpose. Then Reverse Circle Cypher algorithm encrypting the files using random key and finally stored on cloud storage only the cloud controller makes decision about cloud storage files on cloud servers.

VI. CONCLUSION

Duplication of data reduces the performance of the system and also increases the cost of memory utilization. Therefore data de duplication techniques are came to picture which overcome this drawback significantly. The basic idea of this project comes from the fact that cloud is big platform to store and to retrieve the data in huge capacity. Where there is greater possibility of duplication of the data can be happen due to this there will be huge storage space is used unnecessarily. So to avoid this problem proposed system put forwards an idea of avoiding this duplications based on maintaining the hash tags of the files before encryption and conducting advance searches for the existed files using some powerful concepts like bloom filter and inverted index to speed up the complete process.

a.

REFERENCES

[1] Open Security Foundation: DataLossDB (http://datalossdb.org/).
[2] Meister, D., Brinkmann, A.: Multi-level comparison of data deduplication in ī backup scenario. In: SYSTOR '09, New York, NY, USA, ACM (2009) 8:1{8:12
[3] Mandagere, N., Zhou, P., Smith, M.A., Uttamchandani, S.: Demystifying data deduplication. In: Middleware '08, New York, NY, USA, ACM (2008) 12{17
[4] Aronovich, L., Asher, R., Bachmat, E., Bitner, H., Hirsch, M., Klein, S.T.: The design of a similarity based deduplication system. In: SYSTOR '09. (2009) 6:1{6:14
[5] Dutch, M., Freeman, L.: Understanding data de-duplication ratios. SNIA forum (2008) http://www.snia.org/sites/default/files/Understanding_Data_Deduplication_Ratios- 0080718.pdf.

[6] Harnik, D., Margalit, O., Naor, D., Sotnikov, D., Vernik, G.: Estimation of deduplication ratios in large data sets. In: IEEE MSST '12. (april 2012) 1 {11
[7] Harnik, D., Pinkas, B., Shulman-Peleg, A.: Side channels in cloud services: Deduplication in cloud storage. Security Privacy, IEEE 8(6) (nov.-dec. 2010) 40 {47
[8] Halevi, S., Harnik, D., Pinkas, B., Shulman-Peleg, A.: Proofs of ownership in remote storage systems. In: CCS '11, New York, NY, USA, ACM (2011) 491{500
[9] Di Pietro, R., Sorniotti, A.: Boosting e_ciency and security in proof of ownership for deduplication. In: ASIACCS '12, New York, NY, USA, ACM (2012) 81{82
[10] Douceur, J.R., Adya, A., Bolosky, W.J., Simon, D., Theimer, M.: Reclaiming space from duplicate _les in a serverless distributed _le system. In: ICDCS '02, Washington, DC, USA, IEEE Computer Society (2002) 617{632
[11] Storer, M.W., Greenan, K., Long, D.D., Miller, E.L.: Secure data deduplication. In: StorageSS '08, New York, NY, USA, ACM (2008) 1{10
[12] Bellare, M., Keelveedhi, S., Ristenpart, T.: Message-locked encryption and secure deduplication. In: Advances in Cryptology{EUROCRYPT 2013. Springer 296{312
[13] Xu, J., Chang, E.C., Zhou, J.: Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In: 8th ACM SIGSAC symposium. 195{206
[14] Bellare, M., Keelveedhi, S., Ristenpart, T.: DupLESS: server-aided encryption for deduplicated storage. In: 22nd USENIX conference on Security. (2013) 179{194

