Secure Data Deduplication and Auditing for Cloud Data Storage

Patil Varsha[1], Prof. K. N. Shedge[2]

[1]*PG Student, Computer Engineering Department, SVIT, Nashik, India*
[2]*Asst. Professor, Computer Engineering Department, SVIT, Nashik, India*

Abstract - *To avoid redundant data storage on cloud data, deduplication is important aspect in cloud storage platform. Cross client data deduplication technique is widely used on cloud storage servers. The MLE scheme supports secure data deduplication checking. Along with the data security data integrity is important aspect. This work aims to provide a framework for secure data deduplication and data auditing. UR-MLE2 scheme helps to check deduplication whereas data auditor is responsible for data integrity checking. To improve the system performance dynamic binary decision tree is used to check data deduplication. Dynamic binary tree updates the tree data as per user modification or deletion of user data. The systemperformance will be evaluated based on execution time.*

***Key Words*: binary decision tree data deduplication, Convergent key, data auditing, message locked encryption**

1. INTRODUCTION

Large amount of data is stored on cloud. Many organization, companies even the common user uses cloud for data storage. End user tries to outsource the data maintenance task to the cloud server. Cloud server takes the responsibility of data storage, space availability for large scale data and data protection. The data backup service is included in cloud data management servers in case of any natural disaster. The cloud provides various benefits to the end user such as: cost saving, scalable service, mobility Opportunities and simplified access convenience. One of the survey report[2] predicted that the cloud storage will be reached to the size of 40 trillion gigabytes in 2020.

In business sectors data redundancy is growing extensively. To protect data, data-backup is one of the solutions. Another aspect is data version. A small change in one or more files may create a huge change in business documentation. To preserve the old data, data versions are created. The one of the challenge in today's life is the management of ever increasing volume of data on cloud. Instead of keeping multiple copies of same data on a cloud server, a single copy can be preserved. The deduplication technique removes the data redundancy and preserves a single copy of data and generates the references of redundant data copies. To make data management scalable, data deduplication is important technique. It reduces the server storage space and also provides efficient bandwidth usage. When data is outsourced to the third party cloud servers, the issues aeries like data security and privacy. To preserve data security and privacy, encryption is important technique proposed in literature. But traditional encryption schemes do not supports the deduplication technique. In the traditional deduplication technique, every user uses his/her own private key to encrypt the data. As key changes the cipher-text generated from a same document also changes. For a single document different cipher-texts are generated using traditional encryption system and hence it is not at all used for deduplication checking mechanism.

To overcome the problem of traditional encryption technique convergent key mechanism is proposed in a literature[21]. This technique encrypts and/or decrypts the data using convergent key. The convergent key is also called as content hashing key. For key generation, Convergent key generation algorithm uses the content hashing technique i.e. key is generated using data content. Using this convergent key cryptosystem technique identical cipher-text are generated based on identical plaintext files. A convergent key may faces 'confirmation of a file attack'.In this attack, attacker can confirm whether the target posses the file with the with some selected plaintext content or not by simply matching the content with the generated hash key value. To avoid such attacks, Message locked encryption is proposed in literature. This technique provides a secure deduplication.

To provide higher security R-MLE and RMLE2 schemes are proposed using randomized tag generation. But these schemes do not provide good support for data deduplication and these systems are computationally expensive. To avoid these problems of R-MLE scheme, R-MLE2 is proposed in literature. The second important problem faced by the cloud server is data integrity checking. As the user data is transferred to the third part cloud server via internet, client has main concern of data integrity along with the data security. Data integrity checking is nothing but the data maintenance at the cloud end and the assurance for data accuracy and consistency. Cloud data storage is susceptible for threats that arise from both inside and outside of the cloud. The second scenario may arise like cloud server do not notify to the end user in case of any data loss to simply maintain their reputation. To check the data integrity, a system is required that automatically check the data availability and its accuracy without keeping the original copy replica.

Many services provided by the cloud to the third parties which have some limitations such as, data duplication, integrity verification, problem during data transmission etc. There are main two problems in cloud computing such as, data deduplication and data integrity check due to rapid growth of outsourcing data to the cloud are focused in our work. The proposed work present a solution for secure data deduplication and data auditing. For Secure data deduplication URMLE-2 scheme is proposed. To improve the deduplication testing efficiency dynamic deduplication decision tree is used.

1.1 LITERATURE SURVEY

Lot of cloud system studied in this work. Google cloud storage, Drop-Box are widely used cloud applications. These applications also support the deduplication technique. Google simply preserves different version of files rather than keeping bunch of duplicate files.

Some cloud storage systems uses distributed file storage technique. Mark W. Storer Kevin Greenan Darrell D. E. Long Ethan L. Miller[6] proposes a technique for single storage as well as and distributed file storage.

Common user preserves his/her data on mobile devices. Due to space limitation of such handheld devices user prefer to outsource and sync the data on cloud storage such as Microsoft cloud or icloud. Luis Marques and Carlos J. Costa[7] proposes a solution for data deduplication for handheld devices cloud data.

A block level data deduplication is proposed in[8]. In this technique along with the file level deduplication checking block level deduplication is also checked. The file is divided in number of blocks. If no file level deduplication found then block level deduplication is checked. A convergent key mechanism is used to check deduplication over encrypted data. For each data block a new convergent key is generated. Lots of keys are generated for single file. The process of deduplication checking, file upload and download is quit time consuming.

To distribute the deduplication checking and data uploading and downloading the work and to improve efficiency, distribution is done using distributed cloud environment [9]. In this technique hybrid cloud architecture is proposed in which public cloud and private cloud concepts are introduced. This resolves the key management and access privilege problems.

Based on the convergent key technique and block level deduplication checking, Jin Li, Xiaofeng Chen [10] proposes a convergent key deduplication along with key management. Large numbers of convergent keys are generated for cloud data. This technique provides an efficient and reliable key management solution. Users do not need to manage any data keys. The data keys are outsourced to the multiple servers using Decay technique multiple key shares are generated and distributed among multiple servers for security.

To overcome the problem of convergent key problem a ramp secret sharing scheme is proposed by Jin Li, Xiaofeng Chen.[11] In this technique data is divided in number of blocks. For every block ramp secrets are generated. There is no need to create any key for secret generation. These secrets are distributed on multiple servers to provide better security.

The convergent keys are derived from message itself. Hence there is loss of data security. It leaks sort of information namely the plaintext of two messages are identical or not. To overcome this problem, Message lock encryption[4] scheme is proposed in literature. In this scheme along with the convergent key, a public parameter is used to encrypt the data.

This scheme is composed by 5 polynomial time algorithms. To make MLE scheme more stronger, R-MLE2[5] scheme is proposed. In MLE scheme data security depends on the public parameter. In R-MLE scheme fully random tags and deterministic tags are generated. This technique is based on the entropy-based DDH assumption. In this scheme payload is used to store encryption message. The Encryption is done using randomized encryption scheme based on the common key generator g. Equality testing algorithm is proposed to compare tags generated from 2 cipher text, maps the same plaintext or not. For such matching the equality testing algorithm need to be run for entire dataset. This is highly time consuming and impractical to implement in live cloud storage environment.

To overcome the problem of R-MLE2 scheme,UR-MLE-2[1] scheme is proposed by Tao Jiang, Xiaofeng Chen. This scheme uses binary tree structure to preserve the tags. The tag matching is done using equality testing algorithm over binary decision tree. System traverse the tree based on the tag value and matching can be done. This improves the efficiency of tag matching technique using equality testing algorithm. Dynamic updates in tree are also proposed in this work. These dynamic updates are done based on the new file upload and/or cloud file modification.

Along with the data security, data integrity checking is also important. Secure data auditing and data deduplication checking[12] is proposed in literature. In this technique block level deduplication is proposed. An auditor service provides the data deduplication checking and data integrity checking. This technique uses convergent key mechanism for data deduplication and MR-tree for data integrity checking.

1.2 ANALYSIS AND PROBLEM FORMULATION

Due to large outsource data on cloud, there is redundancycloud storage. This causes cloud space wastage. There is need of a system to avoid redundancy in cloud storage environment and provide data deduplication technique over cloud data. The deduplication system must be efficient and can be able to provide deduplication over encrypted data. Along with the deduplication checking there is need to provide data integrity checking on cloud storage environment.

2. SYSTEM OVERVIEW

The proposed system provides a framework for data deduplication checking and data integrity checking. The data deduplication is checked using UR-MLE2 scheme whereas data auditor is responsible for data integrity checking. This system uses dynamic decision tree to check data deduplication. The dynamic decision tree is based on the selfgeneration tree. It allows data update such as data insertion, deletion and modification. The data deduplication checking uses fully random message-locked encryption scheme with randomized tag is an eight-tuple of polynomial-time. The algorithm (PPGen; KeyGen; Enc; Dec; TreeInit; EQ; Valid; Dedup) run by a client and a deduplication server.

The data integrity checking algorithm (metadataGen ,chalgGen, proofGen, verify) run by a data auditor and a deduplication server.The metadata is genrated using SHA1 algorithm.

Following fig. 1 shows the system architecture.

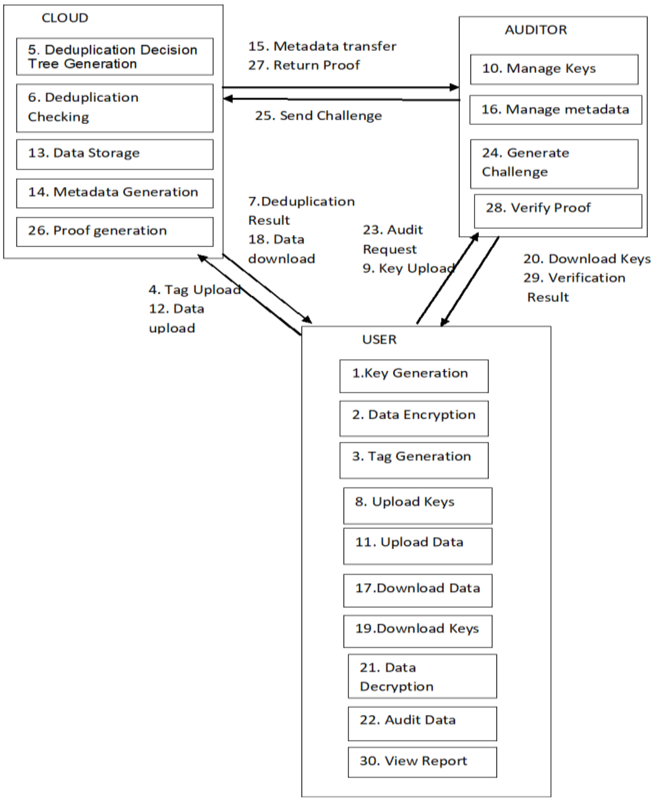

Fig. 1. System Architecture

A. Algorithms

B. R-MLE2 Algorithm

This algorithm is run by a client and a deduplication server. A client does not need to upload its encrypted data item to the storage server when there is a duplicate copy stored. Otherwise, the client needs to upload its data. The server stores a sequence of data fc1,c2,..,cng and the corresponding tag values fT1,T2,..,Tngat some point. The system can efficiently direct to the identical data copy if a duplicate value is stored in the storage server.

Algorithm Steps:

Input: message m,
Output: message derived key km,
Cipertext c,
Decypted Message Dm(equal to m)
Processing:

1) PPGen :genrates the public parameter pp

2) KeyGen: The key generation algorithm takes the public parameters pp and a message m as inputs, and outputs a message-derived key km using SHA-1

3) Enc: The encryption algorithm takes the public parameters pp and the message derived key km as inputs, and returns a ciphertext c using AES-256.

4) TreeInit: outputs the tree state ts of the current database.

5) EQ: takes the public parameter pp and the tags T1 and T2 of two ciphertexts as inputs, and outputs 1 if the tags of the ciphertexts are generated from identical messages.

6) Valid: The validity-testing algorithm takes public parameters pp and the ciphertext c as input. It outputs 1 if the ciphertext c is a valid input and 0 otherwise.

7) Dedup: The data deduplication algorithm takes the public parameters pp, T1, and tag T2 as inputs. It returns whether a duplicate data copy has been found.

8) Dec :The takes the public parameter pp and the message derived key km as inputs. If the algorithm runs successfully, it will return the plaintext Dm using AES 256.

C. EQ: Equality Test Over Dynamic Deduplication Decision Tree The deduplication test is conducted by the storage server.

The dynamic deduplication decision tree is a self-generation tree. The tree construction relies on the self-generated hash value in self-generation tree, and clients can generate the tree paths themselves of the owned data items.

Algorithm Steps:

Input: Tag T

output: Deduplication result :0(Not Found) or 1(Found)
Processing:

1) Client Server : The client asks for the deduplication of

new data m and sends the server T and bi. (Initially, b = -1, which means that the current node is the root of the tree and the corresponding tag is T

2) Server: The server verifies whether

3) Server - Client: The server returns 1 when the equation holds, and 0 otherwise.

4) Client: When the client receives 0 from the server, the client computes s0 Then it computes bi.

5) Client - Server: The client sends bi+1 to the server. Server: The server moves the current pointer over the tree according to bi+1. If bi+1 = 0, the server moves the pointer to its left child. Otherwise, it moves the node to its right child. Then, go to step 3. The algorithm stops when the server receives ”duplication find” or it needs to move the pointer to an empty node.

D. Mathematical Model

System S can be defined as
S= I, O,F
Where
I = fI1, I2, I3g, Set of inputs
I1 = User Authentication details
I2 = Registration details
I3 = User data
O= fO1,O2, O3, O4, O5, O6, O7g Set of outputs
O1 = Encryption Key
O2 = encrypted data
O3 = Decrypted data
O4 = Binary decision tree
O5 = Deduplication result
O6 = Metadata
O7 = Generated Proof
F =f PPGen, KeyGen, Enc, Dec, TreeInit, EQ, Valid, Dedup, metaGen,chalGen, proofGen, verifyg Set of Functions
KeyGen = Generate key
Enc = Encrypt Data
Dec= Decrypt Data
TreeInit = Binary Tree initialization
EQ = equality-testing algorithm
Valid =validity-testing algorithm
Dedup = data deduplication algorithm
MetadataGen = Metadata generation
chalGen = Challenge Message Generation
proofGen = Proof Generation
verify = Verify Proof

3. IMPLEMENTATION

A. Experimental Setup:

For system implementation, Client server architecture is developed. A distributed system is implemented for testing. 3 separate machines are used for implementation. A server side environment is established using apache tomcat-7 and mysql on two system for cloud and auditor. A desktop application is created on third system for client side to communicate with auditor and server. A http Protocol is used to communicate between different entities.

B. Dataset:

A multimedia file dataset is used containing text files, audio, video and images. Compressed files are also used for testing. A synthetic dataset is generated using multimedia files.The sizes of files are ranging from 1mb to 500mb.

C. Performance Metric:

The performance of a system is evaluated in terms of time required for processing for following activities:

1) Deduplication Checking:

2) Data Storage Time

3) Auditing Time

ACKNOWLEDGEMENT

First of all my special thanks to head of Department of Computer Engineering, SVIT, Chincholi, Nashik Prof. Prof. K. N. Shedge, principal Dr. S. N. Shelke for their kind support and suggestions. It would not have been possible without the kind support. We would like to extend our sincere thanks to all the faculty members the department of Computer Engineering for their help. We are also thankful to colleagues for moral support and encouragement. At the end, We are very much thankful to all for direct and indirect help.

REFERENCES

[1] Tao Jiang, Xiaofeng Chen, Qianhong Wu, Jianfeng Ma, Willy Susilo and Wenjing Lou, ”Secure and Efficient Cloud Data Deduplication with Randomized Tag”, in IEEE Transactions on Information Forensics and Security, October 2016, Vol. 12, Issue 3, pp. 532 - 543.

[2] 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-digitaluniverse- in-2020.pdf.

[3] J. Douceur, A. Adya, W. Bolosky, D. Simon, and M. Theimer, ”Reclaiming space from duplicate files in a serverless distributed file system,” in Proc. of IEEE International Conference on Distributed Computing Systems, Macau, China, Jun. 2002, pp. 617-624.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, ”Message-locked encryption and secure deduplication,” in EUROCRYPT

2013, ser. Computer Science, T. Johansson and P. Q. Nguyen, Eds. Springer, 2013, vol. 7881 of LNCS, pp. 296-312.

[5] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev,”Message-locked encryption for lock-dependent messages,” in CRYPTO 2013, ser. Computer Science, R. Canetti and J. A. Garay, Eds. Springer,2013, vol. 8042 of LNCS, pp. 374-391.

[6] M. Storer, K. Greenan, D. Long, and E. Miller, ”Secure data deduplication,”in Proc. of the 4th ACM International Workshop on Storage Security and Survivability, VA, USA, Oct. 2008, pp. 1-10.

[7] L. Marques and C. Costa, ”Secure deduplication on mobile devices,” in Proc. of the 2011 Workshop on Open Source and Design of Communication, Lisboa, Portugal, Jul. 2011, pp. 19-26.

[8] Jin Li,Xiaofeng Chen, Mingqiang Li,Jingwei Li, Patrick P.C. Lee and Wenjing Lou,”Secure Deduplication with Efficient and Reliable Convergent Key Management”, in IEEE Transactions on Parallel and Distributed Systems, Vol. 25, Issue. 6, pp. 1615 - 1625 June 2014.

[9] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, ”A hybrid cloud approach for secure authorized deduplication,” IEEE Transactions on Parallel and Distributed Systems, vol. PP, pp. 1-12, Apr. 2014.

[10] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, ”Secure deduplication with efficient and reliable convergent key management,” IEEE Transactions on Parallel and Distributed Systems, vol. 25, pp. 1615-1625, Nov.
2013.

[11] Jin Li, Xiaofeng Chen, Xinyi Huang,Shaohua Tang,Yang Xiang, Mohammad Mehedi Hassan, Abdulhameed Alelaiwi, ”Secure Distributed Deduplication Systems with Improved Reliability”, in IEEE Transactions on Computers, Vol. 64, Issue. 12, pp. 3569 - 3579, Dec. 1 2015.