Elimination of Equivalent Ciphered Data from Online Based Data Storage

D SivaChidambaram [1], Ramya V [2], Vidya Varshini R [3], Janaki E [4]
[1] Assistant Professor, Department of Computer Science and Engineering Sri Muthukumaran Institute of Technology, Chennai, Tamil Nadu, India.
[2, 3, 4] Student, Department of Computer Science and Engineering Sri Muthukumaran Institute of Technology, Chennai, Tamil Nadu, India.

Abstract – Cloud computing plays a vital role in data storage. In cloud computing data are stored through online, so that it is a trivial process to maintain its privacy and security. As organization increase their reliance on possibly distributed, information system for daily business, they become more vulnerable to security breaches, even as they gain productivity and efficiency advantages Though a number of techniques such as encryption and electronic signature are currently available to protect data when transmitted across sites, a comprehensive approach for data protection must also include mechanism for enforcing access control policy based on data contents. Message Locked Encryption algorithm is discussed about data deduplication in a storage. But this algorithm fails to provide security. To overcome this method, in this paper we propose a scheme called proxy re-encryption. This algorithm works on the situation when the data owners are offline and feels difficult to remove duplicated data. It provides high security and ease of maintenance Both the algorithm are combined together and the result is given.

Index Terms – Cloud computing, data deduplication, proxy re-encrypton, access control.

1. INTRODUCTION

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly. Cloud computing is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment. Cloud computing mainly consists of three major components User, CSP and Third Party Auditor. Cloud offers a vast number of services over the internet one of the main service provided by cloud computing technology is storage. some of the company that provides storage services are amazon azure, sugar sync and carbonate. In cloud the data are stored through online. To maintain data privacy all the data are stored in the form of cyber text the process of converting plaintext into cyber text involves many techniques. Some may not be used due to some of the security issues. Data privacy control is must one to achieve by the cloud service provider because without data privacy and security the data become unsecured. The cloud service is used by numerous number of users, each user uploads data of their own so that there is high possibility that the user can upload a same data more than once. Deduplication is a process of removing the duplicated data in storage which will increase the storage space. The process of removing duplicate repeated data is the must one to achieve. Duplicate data provide a greater problem in the management of data storage because it will occupy unwanted storage and thus more space is waste while backing up the data. Because of maintaining data security all the data are stored in encrypted data so that removal of repeated data becomes impossible, because encrypted data are in cyber formats. There are many algorithms already existing but they are suffering from brute force attack.

The special contribution of this paper is to motivate the users to use cloud storage with data security and privacy. We maintain the deduplication of data in an effective way and maintain the consistency of the database and reduce the space occupied. The remaining portion of this paper is described as follows.

- Define the system and all the security models.
- Provide brief explanation of the scheme.
- Evaluation of security analysis and performance.

2. PROBLEM STATEMENTS

System and model

In this we proposed a algorithm to remove the similar files at Cloud service provider by applying the proxy reencryption and data ownership. It is applicable only at the scenario where the data owners and holders are not online. As above mentioned there are three things in cloud

- User: users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations.
- Cloud Service Provider (CSP): a CSP, who has significant resources and expertise in building and

managing distributed cloud storage servers, owns and operates live Cloud Computing systems.

- Third Party Auditor (TPA): an optional TPA, who has expertise and capabilities that users may not have, is trusted to assess and expose risk of cloud storage services on behalf of the users upon request.

Basically in theoretical wise there is a possibility that CSP and its users can work together but practically it is such collaboration of CSP and user will make the CSP to lose its reputation due to some of the possibility of data leakages. This type of negative impact on the CSP will make the CSP to lose its customers and even make the lose profit. And also the users lose their convenience of storing their data online therefore those collaborations will make the both of them to achieve non profitable business.To make this concept work we have some of the assumption they are:

1. The data holders must provide the correct encrypted hash code
2. User, CSP and AP communicate with each other in a secure channel
3. CSP can authenticate user in order to data storage.
4. User policy must provide to Cloud service provider while getting into the registration process.

Basics and notation

A Proxy Re-encryption scheme is expressed as a tuple of polynomial time algorithms which is encrypt and decrypt the data using the standardized key generation. During the time of input the security parameter 1N, NG provides both a pair of both public and private key (pkA; ska) for entity O. During input pkA and file F, E provides a encrypted cipher text CA = E(pkA;F). During input skA and encrypted cipher text CA, D outputs the plain data F = D(skA;CA). On input (pkA; skA; pkB), the re-encryption key generation algorithm RG, outputs re-encryption key rkA-->B for a proxy. At the input rkA B and encrypted cipher text CA, the re-encryption function R, outputs R(rkA!B;CA) = E(pkB;m) = CB which will be decrypted using a private key skB.

3. WORKING PROCESS

This algorithm involves the following things:

Inserting a data:

In this section the server checks for the duplicate data. If the result is negative, the data holder starts the encryption process with the help of randomly selected symmetric key DEK this type of encryption ensure the security and privacy it also saves the data at the Cloud Service Provider with a token which helps for checking encrypted data. The data owner encrypts DEK with pkAP and passes the encrypted key to CSP.

Removing of duplicate data:

The process of removing duplicate data happens only when the data holder attempts to upload the same file again .The CSP checks all these things with the help of the tokens created. If the result of the token comparison is positive, CSP communicates the authenticated Person for the process of removing the duplicate data by providing the data holder’s PRE public key and it’s token.

Deleting the Data.

When data owners erases file from Cloud Service Provider, CSP at the beginning look at the records of duplicated data holders by deleting theduplication record of this client. If the remaining file is not empty, the Cloud Service Provider does not erase the stored cipher data, but restrict data accessing properties from the holder who actually needs deletion of data. If the remaining file is null, the cipher data must be erased at Cloud Service Provider.

Managing Data Ownership:

For a case a record owner inserts a file after the record holder, the Cloud Service Provider can able to store the file encrypted by the real data owner using DEK which is created by holder Authenticated Person helps re-encryption of Dynamic Encryption Key at Cloud Service Provider for selected data holders.

Updating of Encrypted Data:

For in case that DEK which is reentered by a record owner with DEK and the newly emerged encrypted raw record is given to Cloud Service Provider to change the old storage with the new storage for the purpose of attaining better security and privacy, Cloud Service Provider provides the newly re-encrypted DEK to every data holders with the help of Authenticated Person.

4. PROCEDURES

Removing of Duplicate Data

The process removing of duplicate data at Cloud Service Provider by the help of Authenticated Person using the proposed algorithm. Suppose client U1 stores its valuable record F at Cloud Service Provider with security using DEK1, while the client U2 is a record holder who also attempts to store the same record at Cloud Service Provider. The detailed process of removing duplicate data is presented below:

Step 1 – Generation of Record Token: Client u1 generates record token of F, x1= H(H(F)×p) and sends {x1,pk,Cert(pk1)} to Cloud Service Provider. *Step 2* – Checking of Duplication: Cloud Service Provider checks Cert(pk1) and verify if the duplicated data is saved by finding if x1 really exists. If the verification result is negative, it asks for record upload. Client u1 encrypts record F with DEK1 to get CT1 and encrypted DEK1 with pkAP to get CK1. U1 sends CT1 and CK1 to CSP, which stores them together with x1 and pk1. If the verification is positive and the pre-saved record is from the same client, it informs the client about this situation.

Step 3 – Checking and uploading Encrypted Data: Client u2 later on tries to store the same file or record F at Cloud Service Provider following the same procedure of Step 1 and 2. That is, u2 sends the record package{x2,pk2,Cert(pk2)} to CSP. Duplication happens because x2 exists, so CSP forwards{x2,pk2,Cert(pk2)) to AP.

Step 4 – Ownership challenge: AP challenges the data ownership of u2 by randomly choosingc %%r(0; . . .2-1} and sending it to u2.Theu2 checks c to make sure that 0 < c<2-1,computesy = H(M)+ (s2*c) and sends E(pkap,y) to AP. Ap gets y, computes H(yP+ cV2) and compares it with x2. If H(yP+ cV2)= x2, i.e., the ownership challenge is successful, AP generates re-encryption key rkAP>u2by calling RG(pkAP;skAP; pk2) if it has not been generated and issued to CSP.

Step 5 –Deduplication: CSP re-encrypts E(pkAP;DEK2) by calling R(rkAP_ u2 ;E(pkAP ;DEK1))= E(pk2;DEK1) and provides the re encrypted key E(pk2;DEK1) to u2. Then u2 can get DEK1 with its secret key sk2. The u2 confirms the success of data deduplication to CSP that records corresponding deduplication information in the system after getting this notification. At this moment, both u1 and u2 can access the same data M saved at CSP. User u1 uses DEK1 directly, while u2 gets to know DEK1 by calling D(sk2;E(pk2;DEK1)).

Deletion of Data:

At Cloud Service Provider When record holder U2 needs to delete the record from Cloud Service Provider, it propels removing request to Cloud Service Provider: Cert(pk2), X2. The Cloud Service Provider checks the consistency of the request, then deletes deduplication record of U2, and part U2's later access to F. Cloud Service Provider again checks if the deduplication record is empty. If it is positive, it erases encrypted data CT and the similar data.

Managing Data Ownership:

For a case the real record owner U1 saves the record after the record holder U2, Cloud Service Provider can accomplish to store the record encrypted by the real record owner at the cloud and permit it to share the records and files. The real record ownership is checked after challenging, e.g., the record owner should deliver a specific certificate to display its ownership In this case, Cloud Service Provider communicates with Authenticated Provider by providing all record holders pki (e.g., pk2) if Cloud Service Provider does not know its equivalent re-encryption key rkAP□ui(e.g., rkAP□u2). Authenticated Person issues rkAP□ui to Cloud Service Provider if ownership challenge result is certifiably positive. Cloud Service Provider re-encrypts CK1, acquires re-encrypted DEK1 (e.g., E(pk2;DEK1)), directs it to every related record holders (e.g., u2), erases CT$_2$ and CK$_2$by swapping it with u1's encrypted copy CT1 and CK1, and the apprises equivalent deduplication records.

Encrypted Data Update:

In some cases, a record holder may apprise encrypted cipher record saved at Cloud Service Provider by creating a new DEK0 and enter the newly encrypted record with DEK0 to Cloud Service Provider. The U1 needs to update cipher data saved at CSP with new symmetric encryption key DEK01. Client u1 conducts an update request: {x1; CT1; CK1; update CT1}. Cloud Service Provider saves CT; CK1 together with x1 and pk1. CSP communicates Authenticated Person for removing duplicate data for other record holders if the keys they used are not known. Authenticated Person drafts its policy for creating andsending equivalent re-encryption keys (e.g., rkAP□u2), which are used by CSP to perform re-encryption on CK1 for creating re-encrypted keys that can be decrypted by every eligible record holders (e.g., E(pk2;DEK1)).

5. SIMULATION AND EVALUATING THE PERFORMANCE

As stated above we developed an algorithm to remove the duplicate data in the encrypted data using proxy reencryption and implemented the algorithm. And it worked well in high efficiency. The algorithm cannot get into any problematic behavior while working even with big data and small sized data. In our test we take into account of the following things such as time of data uploading, security level of the each encryption and performance of removing duplicate data procedure.

Testing of efficiency:

Test: encryption and decryption efficiency

In this experiment, the time of uploading the data including the encryption and the time for downloading of data including decryption has been evaluated and by using the proxy re-encryption algorithm. The environment of evaluation having the following properties Intel Core i3 CPU, 4GB of RAM, Dual core Processor, 50GB hard disk. We observed that even if the size of the data is big as 400MB the time required for encryption and decryption is more for big sized data and small sized data. This is an inevitable thing in any encryption algorithm.

The Figure 1 clearly shows the time taken for data uploading using proxy re-encryption and Message Locked Encryption. From this graph we clearly understand that for low memory sized data the uploading time of message locked encryption is slightly smaller than proxy reencryption and if the size increases the taken for MLE is longer than Proxy re-encryption.

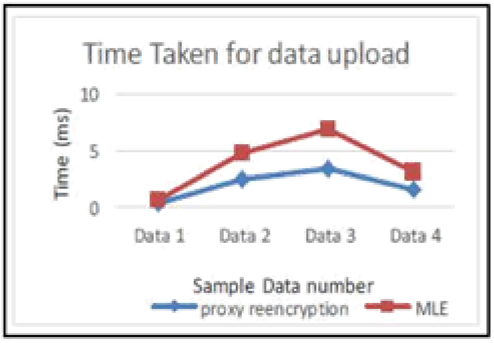

Figure 1.Time taken for uploading data

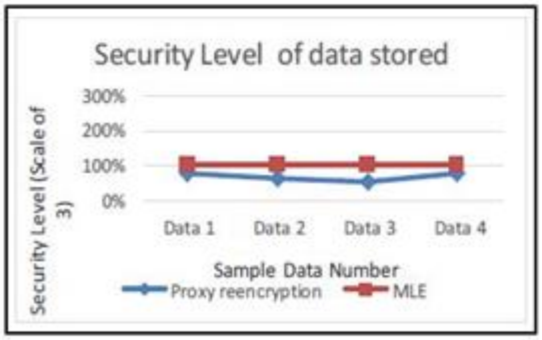

Figure 2.Security level of algorithm

The above graph represent the security level of the two algorithms used while using different sizes of data for proxy re encryption the size of data doesn't matter for any size of data it provides a high security. In case of MLE the security increases only when the size of data increases.

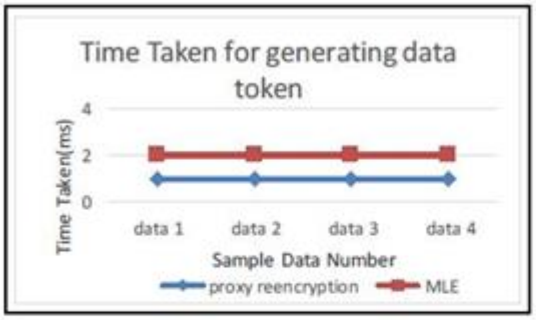

Figure 3.Time taken for providing token

The above graph represents the time taken for the two algorithms to providing token for the data used while using different sizes of data. The time taken is same for both the algorithms.

6. CONCLUSION

In this paper we proposed a scheme for remove duplicate data in the cloud storage which is in the encrypted format using a concept called proxy re-encryption. Finally we succeeded in DE duplicating the encrypted data by achieving it with high data security and privacy.

REFERENCES

[1] BARROSO, L. A., AND HOLZLE, U. The Case for Energy-Proportional Computing. IEEE Computer 40, 12 (December 2007).

[2] ADMINISTRATION, E. I. State Electricity Prices, 2006 [online]. Available from: http://www.eia.doe.gov/neic/rankings/stateelectricitypric e.htm.

[3] BIALECKI, A., CAFARELLA, M., CUTTING, D., AND O'MALLEY, O. Hadoop: a framework for running applications on large clusters built of commodity hardware. Wiki at http://lucene. apache. org/hadoop.

[4] CHANG, F., DEAN, J., GHEMAWAT, S., HSIEH, W.,WALLACH, D., BURROWS, M., CHANDRA, T., FIKES, A., AND GRUBER, R. Bigtable: A distributed storage system for structured data. In Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI'06) (2006).

[5] CHENG, D. PaaS-onomics: A CIO's Guide to using Platform-as-a-Service to Lower Costs of Application Initiatives While Improving the Business Value of IT. Tech. rep., LongJump, 2008.

[6] HAMILTON, J. Cost of Power in Large-Scale Data Centers [online]. November 2008. Available from: http: //perspectives.mvdirona.com/2008/11/28/CostOfPowerIn LargeScaleDataCenters.aspx.

[7] KREBS, B. Amazon: Hey Spammers, Get Off My Cloud! Washington Post (July 2008).

[8] STERN, A. Update From Amazon Regarding Friday's S3 Downtime. CenterNetworks (February 2008). Available from:http://www.centernetworks.com/amazon-s3-downtimeupdate.

[9] Focusing on the Duality of MPL Representation," Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS '07), pp. 57-64, Apr. 2007, doi:10.1109/ SCIS.2007.367670. (Conference proceedings)

[10] VOGELS, W. A Head in the Clouds—The Power of Infrastructure as a Service. In First workshop on Cloud Computing and in Applications (CCA '08) (October 2008)

[11] Nbeel'sBlog,SeenNov2014, http://mohamednabeel.bl ogs pot.ca/2011/03/proxy-re-encryption.html^ "BeSafe - Encrypted Cloud Collaboration". BeSafe. Retrieved 2017-02-07.

[12] M. Blaze, G. Bleumer, M. Strauss. Divertible Protocols and Atomic Proxy Cryptography.

[13] Bertino, E., Sandhu, R. "Database security - concepts, approaches, and challenges." IEEE Transactions on Dependable and Secure Computing 2 (2005): 2-19G.

[14] Ateniese, K. Fu, M. Green, S. Hohenberger. Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage.

[15] Applied Cryptography and Network Security Conference, June 2007.S. Hohenberger, G. Rothblum, a. shelat, and V. Vaikuntanathan.

[16] Securely Obfuscating Re-encryption. Proceedings of the Theory of Cryptography Conference (TCC), 2007.