Survey of De-Duplication on cloud Using Re-Encryption based key management and Convergent Encryption Mechanism

Anuja Phapale
Pune, India

Abstract-*Data outsourcing to the cloud is profitable for reasons of scalability, economy and accessibility but the challenges in security still remain. Cloud computing has many strong economic advantages, but clients are reluctant to trust a third-party cloud provider. To confront these security concerns, data can be transmitted and stored in encrypted form. There are challenges regarding the conditions of the generation, distribution and usage of encryption keys in cloud systems, such as the safe place of keys. As the business organizations move more to the cloud environment the data and therefore the load on the cloud will keep on increasing; therefore an effective storage mechanism is needed so that the data redundancy is reduced. By storing a unique copy of duplicate data, cloud providers greatly reduce their storage and data transfer costs. The advantages of de duplication unfortunately come with a high cost in terms of new security and privacy challenges. To provide ways to improve data security on the cloud and to reduce data redundancy on the cloud is by implementing data de duplication mechanism on cloud. A model for key management based principle of dynamic data re-encryption is practiced to a cloud computing system in a unique way to address the demands of a cloud environment. Furthermore a data de-duplication mechanism is added in order to allow efficient storage in cloud scenarios.*

Keywords-Re-encryption, de-duplication, Cloud security, Cloud computing

I. INTRODUCTION

Data outsourcing to the cloud are profitable for reasons of scalability, economy, and accessibility but important technical challenges still remain. Many modifications to attribute-based encryption [1] are done to allow authorized users access to cloud data based on the required attributes so that the higher processing load from cryptographic processes is assigned to the cloud provider. Its cost effective pay per use model generally results in a small part of the cost of deploying the computing resources in-house.

The amount of data is increasing exponentially by each passing minute, therefore an effective storage mechanism is needed so that the data redundancy is reduced. As the business organizations move more to the cloud environment the data on the cloud will keep on increasing hence duplication of data is certainly undesired. Another important requirement is for data to be accessible with fine grained controls, to provide flexibility. A single user log in is largely deficient in today's data retrieval tasks.

II. LITERATURE SURVEY

Data exchange on the cloud has many solutions such that the cloud provider is not directly trusted. But some solutions are difficult to scale at a large level. For example, the RSA algorithm [2] is dependent upon the factoring of large numbers. The user is given control over on the attribute level [7] but requires a trusted authority and restricts the owner with a pairing operation that is costly in the sense of computation.

Additionally, various proxy re-encryption schemes have been used for storage security. One method is implemented by re encrypting the stored content at the time of retrieval. Such technique can only be applied to an encrypted storage system in which the data owner implements a block encryption mechanism and the keys used in this mechanism are further encrypted to form a lockbox [9].

Proxy re-encryption is sometimes combined with CP-ABE [10] such that cloud provider forms the re-encryption keys based on a pre-shared secret between the data owner and the provider. Another related work suggests the combination of ABE and proxy re-encryption which allows fine-grained access control of resources while giving the responsibility of re-encryption to the cloud provider [11]. To avoid single point of failure, a multi-authority system has also been proposed [12]. Some approaches also require a trusted proxy for each decryption [13] but at the cost of increased communication overhead.

With the potentially infinite storage space offered by cloud providers, users tend to use as much space as they can and vendors constantly look for techniques aimed to minimize redundant data and maximize space savings. A technique which has been widely adopted is cross-user 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 data which is already stored, the cloud provider will add the user to the owner list. Along with low ownership costs and flexibility, users require the protection of their data and confidentiality through encryption. Unfortunately, de-duplication and encryption are two conflicting technologies. While the aim of de-duplication is to detect identical data segments and store them only once, the result of encryption is to make two identical data segments indistinguishable after being encrypted. This means that if data are encrypted by users in a standard way, the cloud storage provider cannot apply de-duplication since two identical data segments will be different after encryption. On the other hand, if data are not encrypted by users, confidentiality cannot be guaranteed and data are not protected against curious cloud storage providers [14].

III. DE-DUPLICATION USING HASH ALGORITHM

A protocol of outsourcing data storage to a cloud in secure fashion is provided. The provider is inadequate to read saved data; authorized users may do so without arbitration by the data owner. An improvement is made over a traditional attribute based encryption model, so that responsibility over key generation is split between a data owner and a trusted authority, the user is freed of the highest computational burdens. Additional security is given through a group keying mechanism, the owner controls access based on the distribution of an additional secret key, beyond possession of the required attributes. Additional secret key is calculated by considering name of the file requested by the user, user's session ID, user ID. This additional security measure is an optional variant applicable to sensitive data accessed frequently.

User will be able to read all the data which is uploaded on the cloud server by the data owner but in order to download the data, owner's permission will be required. This mechanism helps in solving the problem caused due to the unavailability of the data owner for a large extent.

De-duplication helps to remove uploading of same files to the server; it automatically rejects all the similar files. It uses hashing algorithms to check for the replications of the files. It works on the block level i.e. it divides the contents of the files in blocks of data and then uploads or rejects the files depending upon the result of matching the output of hash function with hash functions of previous files. Figure 1 describes the working of hashing algorithm.

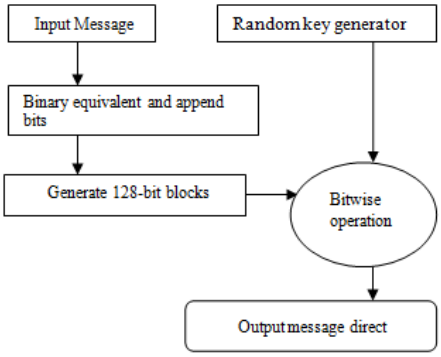

Figure 1: Hashing algorithm

Availability of data owner is a concern because whenever a user requests for the data, owner has to grant the access explicitly. Due to this a lot of time is wasted and prolonged unavailability of owner will almost entirely block the access for the user. To overcome this issue data owner's involvement is restricted for highly sensitive data only.

The hashing function is an integral part of the system and all the outputs of the hash function are stored in a hash table for future use. Every time a new file is to be uploaded all the hash functions are matched with the hash function of the new file in order to check for duplication of data.

The working of the entire system is divided into 6 steps:

Step 1: In the first step, data owner uploads the data on the server and assigns read and download rights to the uploaded files. The data owner also assigns a secret key to each file which is used later as OTP for client's access.

Step 2: The de-duplication mechanism employed then converts the uploaded file into a hash function and matches it with previously generated hash functions in order to remove data redundancy.

Step 3: User has to first register on the system and then log into the system. At the time of registration user's credentials will be stored on the server for future use. User requests for the desired data and if it is not restricted for downloading, then the user can download it. If the requested files are restricted for downloading then data owner's permission will be needed in order to gain complete access. All the user requests have their own requested.

Step 4: For the restricted files data owner will receive the read or write requests from the user along with their respective request IDs and then the owner will grant read or download permission for the requested files. The requests can be viewed by the data owner and request contains request ID, name of

the requestor, names of the files requested for. One data owner can view requests for the files which are self-uploaded and not by some other owner.

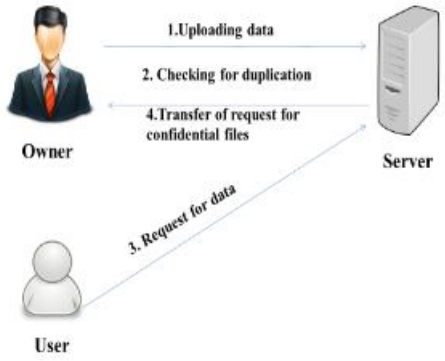

Figure 2: Working of system (steps 1-4)

Step 5: Once the data owner accepts the requests, all the users who requested for read and download access will obtain an OTP through E-mail.

Step 6: After entering the correct OTP user will have read or download access as per requested.

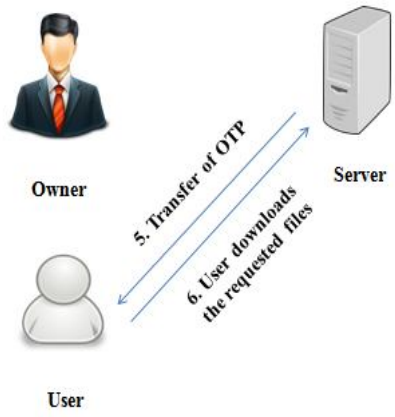

Figure 3: Working of system (steps 5-7)

IV. DATA CONFIDETIALITY USING CONVERGENT ENCRYPTION

The aim of de duplication is to detect identical data segments and store them only once, the result of encryption is to make two identical data segments indistinguishable after being encrypted. If we apply encryption same data, the result will be different.

To meet these two conflicting requirements is convergent encryption whereby the encryption key is usually the result of the hash of the data segment. Although convergent encryption seems to be a good candidate to achieve confidentiality and de-duplication at the same time, it unfortunately suffers from various well-known weaknesses including dictionary attacks: an attacker who is able to guess or predict a file can easily derive the potential encryption key and verify whether the file is already stored at the cloud storage provider or not [14].

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 cipher texts since the encryption key is the same; hence the cloud storage provider will be able to perform de duplication on such cipher texts. 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.

V. CONCLUSION

It is basically a key management system which increases the security of the cloud by providing better access control. Only authorized users are allowed to access the files on the cloud. Data owner's involvement in each and every data transfer is reduced but owner's control over important data is still intact. System can be used in business organizations, colleges, hospitals and all other places where important data is stored on cloud platform. Issue with the availability of data owner has been reduced by further refining owner's control over the data on the server.

De-duplication mechanism reduces the data redundancy on the server and hence provides efficient use of data storage. This mechanism works on block level and hence files with similar names but different content can also be uploaded on the server.

The system is scalable and can be used in almost any cloud environment with a power of growing in terms of security by various modern and better encryption techniques or multi-level authentication techniques as well.

REFERENCES

[1] P.K. Tysowski and M.A. Hasan, "Hybrid Attribute-Based Encryp-tion and Re-Encryption for Scalable Mobile Applications in Clouds," Technical Report 13, Centre for Applied Cryptographic Research (CACR), Univ. of Waterloo, 2013.

[2] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,"Comm. ACM, vol. 26, no. 1, pp. 96-99, Jan. 1983.

[3] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications," Proc. Ninth ACM SIGCOMM Conf. Internet Measurement Conf. (IMC '09),pp. 280-293, 2009.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption,"Proc. IEEE Symp. Security and Privacy (SP '07),pp. 321-334, 2007.

[5] Tassanaviboon and G. Gong, "OAuth and ABE Based Authorization in Semi-Trusted Cloud Computing: Aauth,"Proc. Second Int'l Workshop Data Intensive Computing in the Clouds (DataCloud-SC '11),pp. 41-50, 2011.

[6] X. Liang, R. Lu, and X. Lin, "Ciphertext Policy Attribute Based Encryption with Efficient Revocation," Technical Report BBCR, Univ. of Waterloo, 2011.

[7] J. Hur and D.K. Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems,"IEEE Trans. Parallel and Distributed Systems,vol. 22, no. 7, pp. 1214 1221, July 2011.

[8] Prof. Rakesh Mohanty, Niharjyoti Sarangi, Sukant Kumar Bishi, "A Secured Cryptographic Hashing Algorithm" VSSUT, Burla, Orissa, India

[9] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, pp. 1-30, Feb. 2006

[10] Q. Liu, G. Wang, and J. Wu, "Clock-Based Proxy Re-Encryption Scheme in Unreliable Clouds," Proc. 41st Int'l Conf. Parallel Processing Workshops (ICPPW), pp. 304-305, Sept. 2012.

[11] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM '10, pp. 534-542, 2010.

[12] K. Yang and X. Jia, "Attributed-Based Access Control for MultiAuthority Systems in Cloud Storage," Proc. IEEE 32nd Int'l Conf. Distributed Computing Systems (ICDCS), pp. 536-545, 2012.

[13] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-Based Access Control in Social Networks with Efficient Revocation," Proc. Sixth ACM Symp. Information, Computer and Comm. Security (ASIACCS '11), pp. 411-415, 2011.

[14] Pasquale Puzio, Refik Molva, Melek O¨ nen, Melek O¨ nen, "ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage".