International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Review on Data Sharing Across Cloud Storage Using Key Aggregate Cryptosystem

Rashmi Khawale[1], Omprakash Tembhurne[2]

[1]Department of Computer Engineering, Dr.D.Y.Patil School of Engineering & Technology, University of Pune, India

[2]Professors, Department of Computer Engineering, Dr.D.Y.Patil School of Engineering & Technology, University of Pune, India

Abstract: ***Data sharing is playing vital role in the cloud storage. Using cloud storage user can store and share their data very securely and efficiently. So data access security becomes the critical section to be focused. In this article, we describe a technique which works on how a user can share their data partially over cloud storages. This technique introduces a special type of encryption called as key-aggregate cryptosystem which produces constant size ciphertext. In this technique user can provide a constant-size aggregate key for different choices of ciphertext classes in cloud storage, but the other encrypted files outside the class remain confidential. We implemented this cryptosystem for public-key patient-controlled encryption for flexible hierarchy.***

Keywords: Cloud Computing, Key-aggregate encryption, patient-controlled encryption, leakage resilient cryptosystem.

1. Introduction

The use of cloud computing has increased rapidly in many organizations. Cloud-based services include Software-as-a-Service (SaaS) and Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Cloud computing gives us various facilities for data storage and data sharing. User generally deploys their data over cloud storage in terms of GB or TB. Thus cloud computing is advantageous in terms of low cost and accessibility of data. Ensuring the security of cloud storage is a major factor in the cloud computing environment, as sometime users store sensitive information with cloud storage.

While data sharing in cloud computing environment, data from different users can be stored on separate virtual machines (VMs) but belongs to a single physical machine. But data in a target VM could be stolen by instantiating another VM on same physical machine. Thus considering traditional ways of data privacy, some depends on the server to enforce the access control after authentication [3] or some allows a third-party auditor to check the availability of files on behalf of the data owner without leaking the data [2]. But cloud user can not fully depend on cloud server for their data security and confidentiality purpose. Thus users are motivated to encrypt their data with own keys thus providing access to only desired Receivers.

Let us consider an example, user A uploads a set of photos over cloud. But he does not want to share all these photos with everyone. So he need to put some security constraints. With the available cloud security services user A is not satisfied. So he encrypts his photos using his own keys before uploading. Now when user B asks user A to share his photos, user A will send him a single constant size decryption key via secure channel. With this decryption key, user B is allowed to decrypt only those photos which are permitted by user A. Thus here we can give problem statement as,

"*Design a public key encryption scheme in such a way that any subset of the ciphertext is decryptable by a constant size decryption key*."

The solution for this problem is provided using Key Aggregate Cryptosystem (KAC) [1]. With this solution, user A can simply send user B a single aggregate key via a secure e-mail. Then user B can download the encrypted photos from A's cloud storage space and then use this aggregate key to decrypt these encrypted photos. The sizes of ciphertext, public-key, master-secret key, and aggregate key in this KAC schemes are all of constant size.

2. Key-Aggregate Cryptosystem

A key-aggregate encryption system basically includes five algorithmic steps as follows-

The data owner establishes the public system parameter by using **Setup** and generates a public/master-secret key pair by using **KeyGen**. Messages can be encrypted using **Encrypt** by anyone who also decides what ciphertext class is associated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of ciphertext classes by **Extract**. The generated keys can be passed to Receivers securely via secure e-mails. Finally, any user with an aggregate key can decrypt any ciphertext provided that the ciphertext's class is contained in the aggregate key via **Decrypt**.

- Setup(1^{λ} , n): Data owner executes Setup to create an account on an untrusted server. With input as security level parameter 1^{λ} and the number of ciphertext classes n , it outputs the public system parameter param.

- KeyGen: Data owner executes KeyGen to randomly generate a public/master-secret key pair (pk ,msk)
- Encrypt(pk, i, m): Anyone can execute this step who wants to encrypt data with input a public-key pk, an index i denoting the ciphertext class, and a message m, which outputs a ciphertext C.
- Extract(msk, S): Executed by the data owner to handover the decrypting power for a certain set of ciphertext classes to a Receiver . On input the master-secret key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by Ks.
- Decrypt(Ks, S, i, C): executed by a Receiver who received an aggregate key Ks generated by Extract.On input Ks, the set S, an index i denoting the ciphertext class the ciphertext C belongs to, and C, it outputs the decrypted result m if i ϵ S.

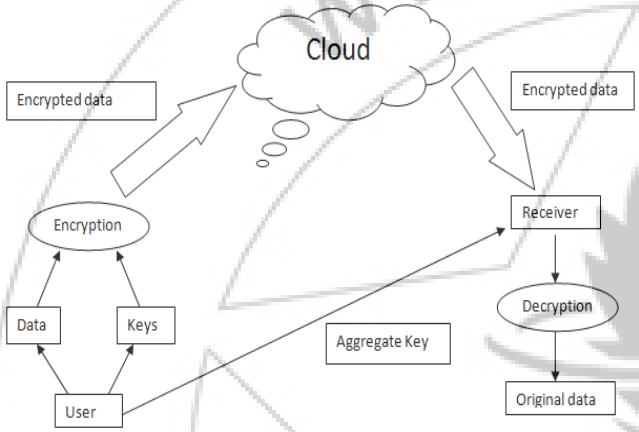

Figure 1:Data flow architecture

Above figure shows how data flows between user and receiver.

3. Related Work

In this section basic KAC scheme is compared with other possible solutions on sharing in secure cloud storage.

a) Cryptographic Keys for a Predefined Hierarchy

Cryptographic key assignment schemes works on the basis of minimize the expense in storing and managing secret keys for general cryptographic use by using a tree structure [5]. By using hierarchical tree structure , a key for a given branch can be used to derive the keys of its descendant nodes.This can solve the problem partially if one intends to share all files under a certain branch in the hierarchy which alternatively means that the number of keys increases with the number of branches. So it is difficult to create a hierarchy that can save the number of total keys to be granted for all individuals simultaneously.

b) Compact Key in Symmetric-Key Encryption

This method is used to generate a secret value instead of a pair of public/ secret keys [6]. It is designed for the symmetric-key setting in which the encryptor gets the corresponding secret keys to encrypt data. Thus it is unclear how to apply this idea for public key encryption scheme.

c) Compact Key in Identity-Based Encryption (IBE)

In this encryption, there is a trusted party called private key generator in IBE which holds a master-secret key and gives a secret key to each user with respect to the user identity. The encryptor can take the public parameter and a user identity to encrypt a message [7]. The receiver can decrypt this ciphertext by his secret key. Some tried to build IBE with key aggregation. But their key-aggregation comes at the expense of O(n) sizes for both ciphertext and the public parameter, where n is the number of secret keys . This greatly increases the costs to store and transmit ciphertext.

d) Attribute-based encryption (ABE)

This scheme maintains each ciphertext to be associated with an attribute, and the master-secret key holder can extract a secret key for a policy of these attributes so that a ciphertext can be decrypted by this key. But the size of the key often increases linearly with the number of attributes it encompasses, or the ciphertext-size is not constant [8].

This all comparison can be summarized in following table

Table 1: Comparison between KAC and other schemes

	Decryption Key Size	Ciphertext size	Encryption Type
Key Assignment schemes for predefined hierarchy	Non constant	constant	Symmetric or public key
Symmetric-key encryption with compact key	constant	constant	Symmetric-Key
IBE with compact key	constant	Non Constant	public-Key
Attribute based Encryption	Non Constant	constant	public-Key
KAC	constant	constant	public-Key

4. Patient-Controlled Encryption (PCE)

We implemented this key aggregate cryptosystem in preserving patient's privacy in electronic health record systems [4]. Moving to electronic health records is important to the modernization of healthcare system. But computerized medical records are vulnerable to cyber attacks. Also patient may need to share their data partially with some users. Thus designing Patient Controlled Encryption (PCE) provides solution to secure and private storage of patients' medical records.

In PCE, the health record is decomposed into a hierarchical tree structure based on the use of different ontologies, and

patient is the one who generate and store secret keys. So whenever there is a need to access part of the record, a patient will release the secret key for the concerned part of the record. Thus any patient can either define his own hierarchy according to his need, or follow the set of categories suggested by the electronic medical record system, such as disease, x-rays, doctors, allergies, medications, and so on. When the patient wishes to give access rights to her doctor, he can choose any subset of these categories and provide a single key, from which keys for all these categories can be computed. Thus, this cryptosystem helps user to securely and partially share the data over cloud.

5. Proposed System

Although KAC provides constant-size keys , when one carries the delegated keys in a mobile device without using special trusted hardware, the key is prompt to leakage, thus designing a leakage-resilient cryptosystem [9] is the proposed work for the system. Leakage-resilient cryptosystem attempts to tackle attacks over its data. Leakage-resilient cryptosystem maintain their security even if an attacker learn some partial information of their data.

6. Conclusion

Thus data privacy and security is maintained by designing a public key cryptosystem called as Key Aggregate Cryptosystem (KAC). This KAC helps user to share their data partially over cloud with constant size key pair of public-master keys and also receiver can decrypt this data with single constant size aggregate key. This helps us to create Patient-Controlled Encryption (PCE) system. There are some limitation to the existing system like predefined bound of the number of maximum ciphertext classes and system is prompt to leakage of key.

References

[1] [Cheng Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng,,” Key Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage “,IEEE Transaction on Parellel and Distributed System, vol. 25, no. 2, February 2014 .
[2] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans.Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
[3] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, “SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment,” Proc. 10th Int’l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.
[4] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, “Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records,” Proc. ACM Workshop Cloud Computing Security (CCSW ’09), pp. 103-114, 2009.
[5] S.G. Akl and P.D. Taylor, “Cryptographic Solution to a Problem of Access Control in a Hierarchy,” ACM Trans. Computer Systems,vol. 1, no. 3, pp. 239-248, 1983.
[6] J. Benaloh, “Key Compression and Its Application to Digital Fingerprinting,” technical report, Microsoft Research, 2009.
[7] F. Guo, Y. Mu, and Z. Chen, “Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key,” Proc. Pairing-Based Cryptography Conf. (Pairing ’07), vol. 4575,pp. 392-406, 2007.
[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,” Proc. 13th ACM Conf. Computer and Comm. Security (CCS ’06),pp. 89-98, 2006.
[9] S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, “Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions,” Proc. ACM Conf. Computer and Comm. Security, pp. 152-161, 2010.

Author Profile

Rashmi Khawale Research Scholar Dr. D.Y.Patil School of Engineering & Technology, Pune, University of Pune. She received B.E. in Computer Engineering from Shri Shivaji Shikshan Society’s College of Engineering and Technology, Akola. Currently She is persuing M.E. in computer engineering from Dr.D.Y.Patil School of Engineering & Technology, Pune, University of Pune.

Prof. Omprakash Tembhurne received the B.E. and MTech degrees in Computer Science Engineering. Currently he is working as Assistant Professor of Computer Engineering Department in Dr. D.Y.Patil School of Engineering & Technology, Pune, India