Volume: 2, Issue: 8, 391-393
Aug 2015
www.allsubjectjournal.com
e-ISSN: 2349-4182
p-ISSN: 2349-5979
Impact Factor: 3.762

M Ramesh
M.Tech Student in CSE,
Chadalawada Ramanamma
Engineering College,
Tirupati, India.

R Suresh
Professor & Head of the
Department of Computer
Science and Engineering,
Chadalawada Ramanamma
Engineering College,
Tirupati, India.

Correspondence
M Ramesh
M.Tech Student in CSE,
Chadalawada Ramanamma
Engineering College,
Tirupati, India.

Compact key cryptographic approach to store & access Distributed stored information from cloud

M Ramesh, R Suresh

Abstract
Day to day cloud expands their IT services in a wide range to increase their potentiality in worldwide, among them data outsourcing is one of the major functionality in cloud storage. Maintaining the security, privacy and effective way of data sharing becomes a critical job for cloud providers. To overcome this problem, we newly presented a symmetric key cryptographic approach which generates a constant size cipher text. An individual user can effectively decrypt the encrypted data from shared cloud. A compact key contains set of secret keys which is generated by the key distribution center on the basis of file and user's privileges who want to access the data from the cloud storages. We can able to store this compact key conveniently in limited stored space and delivered to the intend user through secure channels. We evaluate the functional behavior of security, efficiency, flexibility and key generation process within a Patient-Controlled Encryption application with some authorized users accessing from different geographical locations.

Keywords: Cloud storage, data, distributing, compact key encryption, patient-controlled encryption.

1. Introduction
Cloud storage is nowadays very popular storage system. Cloud storage is storing of data off-site to the physical storage which is maintained by third party. Cloud storage is saving of digital data in logical pool and physical storage spans multiple servers which are managed by third party. The Third party is responsible for keeping data available and accessible and physical environment should be protected and running at all time. Instead of storing data to the hard drive or any other local storage, we save data to remote storage which is accessible from anywhere and anytime. It reduces efforts of carrying physical storage to everywhere. By using cloud storage we can access information from any computer through internet which omitted limitation of accessing information from the same computer where it is stored.
While considering data privacy, we cannot rely on traditional techniques of authentication, because unexpected privilege escalation will expose all data. The Solution is to encrypt data before uploading to the server with user's own key. Data sharing is again important functionality of cloud storage, because user can share data from anywhere and anytime to anyone. For example, an organization may grant permission to access parts of sensitive data to their employees. But challenging task is to share that how encrypted data. The Traditional way is user can download the encrypted data from storage, decrypt that data and send it to share with others, but it loses the importance of cloud storage.
Cryptography technique can be applied in a two major ways- one is a symmetric key encryption and other is asymmetric key encryption. In symmetric key encryption, same keys are used for encryption and decryption. By contrast, in asymmetric key encryption different keys are used, the public key for encryption and private key for decryption. Using asymmetric key encryption is more flexible in our approach. This can be illustrated by following example.
Suppose Alice puts all data on Box.com and she does not want to expose her data to everyone. Due to data leakage possibilities she does not trust on privacy mechanism provided by Box.com, so she encrypt all data before uploading to the server. If Bob asks her to share some data, then Alice use share function of Box.com. But the problem now is to share that how encrypted data. There are two severe ways:

1. Alice encrypts data with single secret key and shares that secret key directly with the Bob.
2. Alice can encrypt data with distinct keys and send Bob corresponding keys to Bob via secure channel.

In the first approach, unwanted data also get exposed to the Bob, which is inadequate. In the second approach, no. of keys is as many as no. of shared files, which may be a hundred or thousand as well as transferring these keys require secure channel and storage space which can be expensive.

Therefore best solution to above problem is Alice encrypts data with distinct public keys, but sends single decryption key of constant size to Bob. Since the decryption key should be sent via secure channel and kept secret small size is always enviable.

2. Literature Review

In latest cryptography area, a fundamental problem we often study is about leveraging the secrecy of a small piece of knowledge into the ability to perform cryptographic functions (e.g. encryption, authentication) several times. In this paper, we study how to create a decryption key more powerful in the sense that it allows decryption of multiple cipher texts, without increasing its size. Specifically, our problem statement is "To design an efficient public-key encryption scheme which supports flexible delegation in the sense that any subset of the cipher texts (produced by the encryption scheme) is decrypt able by a constant-size decryption key generated by the owner of the master-secret key)."

We now solve this problem by introducing a different type of public-key encryption which we call key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public-key, but also under an identifier of cipher text called class. That means the cipher texts are further categorized into various classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for various different classes. Importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of cipher text classes. With our solution, Alice can simply send Bob a single aggregate key via a secure e-mail. Now Bob can download the encrypted photos from Alice's Drop box space and then use the same aggregate key to decrypt these encrypted photos.

3. 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 public/master-secret key pair by using Key Gen. Message scan been crypted using Encrypt by anyone who also decides what cipher text class is associated with the plain text message to been crypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of cipher text 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 cipher text provided that the cipher text's class is contained in the aggregate key via Decrypt.

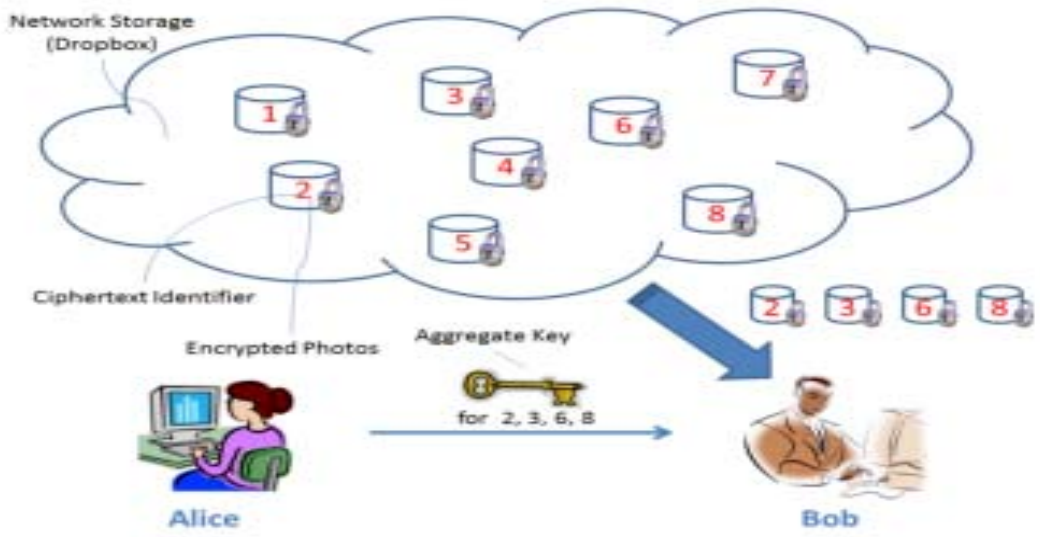
Setup (1λ, n): Data owner executes Setup to create an account on an un-trusted server. Within put as security levelparameter1λandthenumberofciphertextclassesn, it outputs the public system parameter.

Key Gen: Data owner executes Key Gen 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 indexidenoting the cipher text class, and a message m, which out putsa cipher text C.

Extract (msk, S): Executed by the data owner to hand over the decrypting power for a certain set of cipher text 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 cipher text class the cipher text C belongs to, and C, it outputs the decrypted result mif i є S.



4. Related work

KAC scheme is compared with other possible solutions on sharing insecure 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 encryption 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 encryption can take the public parameter and a user identity to encrypta message [7]. The receiver can decrypt this cipher text by his secret key. Some tried to build IBE with key aggregation. But their key-aggregation come sat the expense of O (n) sizes for both cipher text and the public parameter, where n is the number of secret keys. This greatly increases the costs to store and transmit cipher text.

D) Attribute-based encryption (ABE)

This scheme maintains each cipher text 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 cipher text can be decrypted by this key. But the size of the key of ten increases linearly with the number of attributes it encompasses, or the cipher text-size is not constant [8].

This all comparison can be summarized in following table:

	Not constant	Cipher text 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

5. Patient-Controlled Encryption (PCE)

We implemented this key aggregate cryp to system in preserving patient's privacy in electronic health record systems [4]. Moving to electronic health records is important to the modernization of health care system. But computerized medical record sare 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 trees 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 soon. 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 categorie scan be com puted. Thus, this cryptosystem helps user to securely and partially share the data overcloud.

6. Conclusion

Thus data privacy and security is maintained by designing a public key cryp to system 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 key sandal so receiver can decrypt this data with single constant size aggregate key. This helps us to create Patient-Controlled Encryption (PCE) system. There is some limitation to the existing system like predefined bound of the number of maximum cipher text classes and system is prompt to leakage of key.

7. References

1. Cheng Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng,," Key Aggregate Cryptosystem for Scalable Data Sharingin CloudStorage", IEEE Transactionon 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. Akland P.D. Taylor, "Cryptographic Solutiontoa Problem of Access Controlina 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: Howto Decrypt Multiple Ciphertexts Usinga 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.