International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 10 (2015) pp. 26521-26528

http://www.ripublication.com

Secure Delegation Using Key Aggregate Cryptosystem

M.Vinothkumar[1], Dr.N.Balaji[2],
[1]Assistant Professor, Department of CSE, University College of Engineering, Dindigul, Tamilnadu, India
[2]Professor/Head, Department of Information Technology, KLN College of Engineering, Madurai, Tamilnadu, India
E mail Id: vinoth_rajan2002@yahoo.co.in[1], balajin@klnce.edu[2]

Abstract

The most important concept of cloud computing is data sharing. The data must be shared efficiently among users in secure way. Various cryptographic schemes are available to provide the delegation of decryption rights in cloud environment. Here we introduce the concept of aggregate key cryptosystem to facilitate the secure delegation in an efficient and secure manner. The aggregate key introduced in our scheme has the power of decrypting set of cipher texts instead of decrypting single cipher text. Only the files requested by the user can be decrypted using a single aggregate key from the cloud storage directly while the other files remain encrypted. Any subset of cipher texts can be decrypted by a single aggregate key in our scheme. Hence this aggregate key requires less storage space and can be conveniently sent over network.

Index Terms: Cloud computing, data sharing, aggregate key cryptosystem, decryption rights.

Introduction

Cloud storage is the method of storind data online in the cloudso that a company's data is stored in the cloud and can be accessed by multiple persons and resources which is referred as cloud environment.Cloud storagegives us the benefits of accessibility, deployment ,reliability, protection for data backup,archival which leads to lower storage costs are so that the there is no need for purchase, manage and maintain expensive hardware. However, cloud storage does have security and agreement concerns.[12]. Personal cloud storage is a type of public cloud storage that applies to storing the users data in the cloud and providing the individual with access to the data from anywhere in the world. Apple's iCloud is one such example of cloud storage.It also provide data storing and sharing capabilities across various devices in

the network.In private cloud storage, the storage provider has infrastructure in which the company's data center that is maintained by the storage provider. Private cloud storage resolves the potential threats in cloud storage but at the same time providing performance and security.

However, cloud data is suceptible to the similar threats as the websites suffer from.A threat like distributed-denial-of-service can modify the content of all the data. Besides software and hardware failure, even error due to user can also lead to failure in cloud security.

Amazon's EC2 cloud services experienced a major crash. While the data lost is insignificant when matched to the data stored by an enterprise, it may be a threat for data loss in some companies.[5].For example,Chartbeat which is an Amozon's customer, informed their users that the data stored has been permanently deleted. In such cases to work with the sensitive data the private cloud seems to be the better option. Dropbox is one such example for hosting file service.Dropbox is a type of service that allows hosting file in the cloud .Itoffers data services, client software, synchronization of files, personal cloud, and cloudstorage. Dropbox helps users to create a special folder on their devices, which Dropbox then centralize so that it appears to be the same folder regardless of which device is used to view the files. Files can be stored in the folder and accessed whenever needed by the user.Drop box file sharing works in cohesion with file synchronization which keeps the file routinely updated across all shared nodes therefore every single recipient will always receive the latest version of the file.

The challenging problem in this is to safely distribute data among users in the cloud. Security mechanisms alone cannot be relied for protectingdata stored in the cloud since they can be the risk of security breach [2] Moreover each decryption key will have the power of decrypting a single encrypted file. Hence the number of decryption keys increases linearly with the increase in amount of files to be decrypted in case of cloud environment [6].

In the proposed work, these limitations can be overcome by the Key Aggregate Cryptosystem scheme (KAC) where in we use a single decryption key to decrypt multiple files.[1] This aggregate key consumes limited storage compared to conventional decryption method. The aggregate key can be securely transmitted and can be stored within limited storage requirements [3]. Consider DropBox as an example in which Alice wants to share some of the encrypted photos on her cloud storage to Bob. The traditional way of decryption allows sending the decryption keys of requested files. This results in increase in the amount of decryption keys with respect to the number of requested files. Our approach allows for a single decryption key to be sent to decrypt any subset of cipher texts.So that Alice can send a single decryption key for any set of photos requested by Bob.

Related Work

This section surveys the relevant literature work about the concept of reducing the number of keys granted for decryption

D. Boneh, et.al., suggested BroadcastEncryption scheme in which broadcaster encrypts a message only for a specified subset of users S who listen to the broadcasting channel.The user in subset S can use their private key for decrypting the message.The users outside the subset S cannot obtain the content of the broadcast.Public keys in this system increases linearly with the total number of receivers**.**Although their scheme supports constant-size secret keys, every key only has the decrypting power only for ciphertexts associated to a particular index.[11]

V. Goyal, et.al., proposed Key-Policy Attribute-Based Encryption (KP-ABE).Since the data stored in the cloud tends to be more sensitive, there will be a need to encrypt data stored at the cloud. The major drawback of encrypting data, is that it can be shared only at a coarse-grained level. The newly developed cryptosystem allows fine-grained sharing of encrypted data which is called Key-Policy Attribute-Based Encryption. In the cryptosystem, cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. Thus demonstrating the applicability of construction to sharing of audit-log information,broadcast encryption and supports delegation of private keys which subsumes Hierarchical Identity-Based Encryption (HIBE)[4].The main drawback in this cryptosystem is that, ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt.However the major concern in ABE is collusion-resistance but not the compactness of secret keys.[7].

T. Okamoto, et.al., presented Adaptively Secure General Inner-Product Encryption.Two non-zero inner-product encryption (NIPE) schemes that are adaptively secure under a standard assumption were presented. Any previous IBR scheme with constant-size cipher texts or constant-size secret-keys was not adaptively secure in the standard model. Thus two zero inner-product encryption (ZIPE) schemes are presented each of which has constant-size cipher texts or constant-size secret-keys and is adaptively secure under the DLIN assumption in the standard model which imply an identity-based broadcast encryption (IBBE) system with constant-size cipher texts or constant-size secret-keys that is adaptively secure under the DLIN assumption. Thus an identity based revocation system with constant size cipher text that is adaptively secure is proposed. But the size of the key often increases linearly with the number of attributes it encompasses [10].

G. Ateniese, et.al.presented Hierarchical Key Assignment Schemes.The time-bound hierarchical key assignment schemes which are provably-secure and efficient are designed .Both the unconditionally secure and the computationally secure settings are considered.[9]. A time-bound hierarchical key assignment scheme is a method to assign time-dependent encryption keys to a set of classes in a partially ordered hierarchy, so that each class can calculate the keys of all classes lower down in the hierarchy, according to temporal constraints. The first one is based on symmetric encryption schemes, whereas, the second one uses concept of bilinear maps.The scheme assigns cryptographic keys to a set of partially ordered classes so that a cryptographic key of a higher class can be used to derive the cryptographic key of a lower class.But granting the parent key implicitly grants all the keys of its descendant nodes [8].

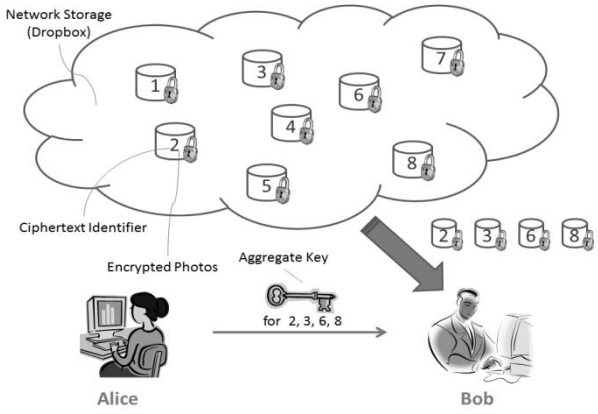

Figure 1: System diagram of the proposed Key Aggregate Cryptosystem model

Key Aggregate Cryptosystem Model

This section explains the detailed description about the proposed Key Aggregate Cryptosystem model.Fig.1 depicts the overall flow diagram of the proposed model.

A. The System Model

This method provides key aggregation to reduce the decryption keys.The proposed system design an efficient encryption scheme whichsupports flexible delegation in the sense that any subset of the cipher texts are decryptable by a constant-size decryption key.The system contains a special type of encryption called key aggregate cryptosystem (KAC).

In KAC, users encrypt a message not only under a symmetric key, but also under an identifier of ciphertext called class.The key owner holds a master-secret referred to as master-secret key, which can be used to mine secret keys for any subset .The users encrypt a message not only under a public-key, but also under a special type identifier of ciphertext which is called class. That means the ciphertexts are additionallycharacterized into different classes.The extracted key produced can be an aggregate key which as compact as a secret key which is similar to the decryption key of a single class, but aggregates the power of any subset of decryption keys, i.e., thedecryption authority for any subset of ciphertext classes.

B. Design Objectives

To facilitate the proposed model the design must comply with the task of reducing the number of decryption keys.

1. Data owner setup phase:

The key aggregation property is especially useful when the delegation must be efficient and reliable. The schemes enable the users to share the data in a confidential and selective way, with a fixed and small ciphertext extension, by giving away to each approved user a single and undersized aggregate key.The user implement setup

algorithm. The setup algorithm takes input such as user id and name of the data owner. The valid user once registered can access the database and store the files into his/her local database.

2. Key Extraction

Key extraction will be executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegatee. Once the valid user wants to encrypt the data he will store the file in a particular index.When the file to be encrypted is selected, the index for the file within which it will be stored is generated automatically.

3. Key generation

Key generation is the process of generating keys in the cryptography. A key is used to encrypt and decrypt any data that is to be encrypted or decrypted. Symmetric-key algorithms use a single shared key; keeping data secret requires keeping this key secret. SHA1 outputs a 160 bit digest of any sized file or input. The construction is similar to the MD4 and MD5 hash functions.

The key generation algorithm takes as input the public key and a set of attributes S that describe the key. The algorithm outputs the secret key and access level of the file that must be allowed. It can be made either public or private.

4. Data Encryption

Data that can be understood and can be read without any special technique is called plaintext or clear-text. The method of making plaintext in such a way to hide the content is called encryption. Encrypting plaintext results in scribbled format called ciphertext. The process of reversing cipher-text to its original plaintext is called decryption.

The encryption algorithm takes as input the public parameters, a message, and an access structure over the universe of attributes. The algorithm will encrypt message and produce a ciphertext such that only a user that possesses a set of attributes and access structure will be able to decrypt the message.

5. Decryption using aggregate key:

Once the data owner obtains the master key for the specified set of files required by the user, he will send only the aggragate key to the user.The user can use this aggregate key for decrypting all the required files directly from the cloud server while the data outside it remains secure.

Performance Analysis

This division evaluates the performance of the proposed system with the parameters of no of keys granted and variation of execution time in symmetric and asymmetric algorithms.

A. No of keys granted in Hierarchical Key assignment vs Key aggregate scheme
The number of keys granted increases linearly with the increase in number of files in case of existing hierarchical key assignment whereas the aggregate key reduces the number of keys granted since it does not confine to any subset.

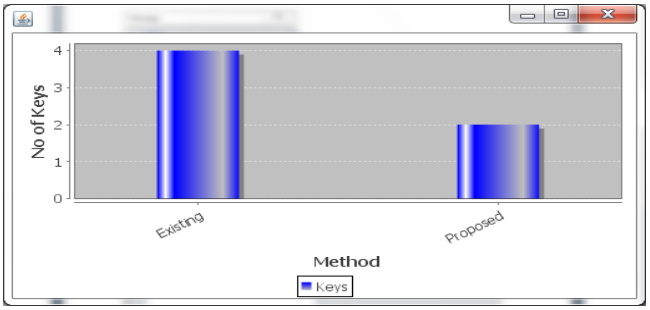

Figure 2: No of keys granted in existing vs proposed system

B.Time Taken for Execution
The time taken for execution in terms of algorithm used is shown in the Fig.3. The time taken for symmetric algorithms has lower execution time than the time taken for asymmetric algorithms.

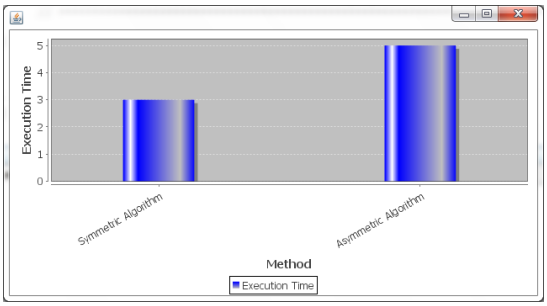

Figure 3: Time taken for execution by symmetric and asymmetric algorithms

Conclusion and Future Work

This paper proposes the key aggregation scheme which is helpful in reducing the amount of decryption keys needed.Protecting user's data privacy is a central question of cloud storage.Cryptographic techniques are getting more flexible and often involve multiple keys for a single application. Compressing secret keys in key aggregate cryptosystems support delegation of secret keys for different ciphertext classes. Thus the approach is easier than hierarchical key assignment which can only save spaces if all key-holders share a similar set of subset.

In future enhancement the predefined bound on the number of maximum ciphertext classes should be eliminated. Since the number of ciphertexts usually grows rapidly, the solution is to reserve enough ciphertext classes for the future extension.

References

[1] Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng, Senior Member, IEEE.IEEE Transactions on Parallel and Distributed Systems. Volume: 25, Issue: 2. Year: 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] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '03), pp. 416-432, 2003.

[4] M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies,"ACM Trans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43, 2009.

[5] 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

[6] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," Proc. Information Security and Cryptology (Inscrypt '07), vol. 4990, pp. 384-398, 2007.

[7] 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

[8] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 1, pp. 182-188, Jan./Feb. 2002.

[9] G. Ateniese, A.D. Santis, A.L. Ferrara, and B. Masucci, "Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," J. Cryptology, vol. 25, no. 2, pp. 243-270, 2012

[10] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption," Proc. 10th Int'l Conf. Cryptology and Network Security (CANS '11), pp. 138-159, 2011.

[11] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys,"Proc. Advances in Cryptology Conf. (CRYPTO '05), vol. 3621, pp. 258-275, 2005.

[12] 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