SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC TRANSACTIONS ON NETWORKS AND COMMUNICATIONS

VOLUME 5, NO. 2
ISSN: 2054 -7420

Achieving Scalability with Data Owner Anonymity in Cloud Access Control

Abdulqader A. Bahaj, and Ahmed M. Abouollo
King Fahd University of Petroleum & Minerals, Computer Networks, Saudia Arabia
abdulqader.bahaj@gmail.com

ABSTRACT

Cloud computing is a trending technology that enables subscribing organizations to outsource computations and storage, and eliminates the need of purchasing and maintaining the equipment by the organizations themselves. However, it is very challenging to maintain the privacy and security of data especially when the number of users grows dramatically. This paper focuses on achieving a high level of scalability to the cloud, allowing fine-grained access control, preserving the anonymity of the data owner and enabling the end user to verify the integrity of the data uploaded to the cloud. In order to achieve this, this paper proposes an effective scheme that uses Ciphertext Policy Attribute Based Encryption (CP-ABE) combined with identity-based encryption (IBE), and introduces a security mediator which signs files on behalf of the data owner to preserve the data owner's anonymity from the cloud. This scheme allows the end user to check the integrity of the data on the cloud.

1 Introduction

Cloud computing paradigm has been and will continue to be one of the most effective techniques to outsource computation and storage. Despite the great advantage cloud computing provides, it suffers from data security and privacy risks. Since the outsourced data is usually out of the trusted domain servers, counter precautions need to be taken to avoid any external or internal risk such as unauthorized access to data by users who obtain more access than they should be granted, intruders who gain access to the system without being given permission to start with, or even the cloud administrators getting paid to leak user data.

Every user of a system has a set of privileges that might not necessarily be appropriate for others. Well defined access to each piece of data must be managed for each user to ensure privacy and security. A possible technique to target this in cloud computing is to encrypt data and share keys with the privileged users accordingly. This technique becomes a hassle when the number of cloud users booms and when fine-grained data access is needed.

Therefore, it is crucial to find a solution that achieves high level of scalability while maintaining the security of data and the privacy of cloud users.

DOI: 10.14738/tnc.52.2764
Publication Date: 30th March 2017
URL: http://dx.doi.org/10.14738/tnc.52.2764

2 Problem Statement

This paper targets preserving the privacy and anonymity of the data owner in a cloud environment's access control scheme. The scheme must be highly scalable, provides fine-grained access and enables data users to verify the integrity of the data uploaded to the cloud.

3 Background and Terminology

Access control is concerned with regulating who can view, edit and use resources in a computing environment. Most customers require the cloud environment to support fine grained access control, which specifies precisely in a good level of details the characteristics of whoever is granted access, the properties to grant access to, and the level of authority for each user.

Anonymity of data owner in a cloud environment is sometimes crucial. This property can be achieved by introducing a security mediator (SEM), which is a server that is responsible for generating the signature on outsourced data on behalf of the data owner. However, this SEM is not supposed to have visibility to the data that needs to be signed by itself. Therefore, blinding techniques are needed to be introduced. Blinding techniques are meant to allow an agent to provide a service such as signing messages to a client in an encoded form without being able to see the real input or output.

4 Related work

There are many schemes that can be used to encrypt data over a cloud environment. Some of them are more efficient than the others. It should be noted that there are two main points that need to be provided by each scheme to be qualified as a potential scalable choice: Fine-Grained Access and Key Delegation. The following is a presentation of some of the possible access control schemes. It is noteworthy here that none of these schemes provides anonymity of the data owner except the Traditional Symmetric Key cryptosystem, since it uses one shared key between several users.

4.1 Traditional Symmetric Key Cryptosystem Scheme

In the Traditional Symmetric Key scheme pointed out in [3], the sender uses one shared key with all the recipients to encrypt a file and store it in the cloud so that the recipients are able to get the encrypted file from the cloud and decrypt it with the shared key.

This scheme does not qualify to be scalable according to the two points we mentioned earlier. It doesn't provide fine-grained access to the encrypted files as every authorized user has the same shared key, as well as it does not support key delegation.

It is worth mentioning that if the data owner wishes to stop access to a certain file for a certain user who was granted access before, he needs to change the key, re-encrypt the file and re-distribute the key to all other users. As such, the Traditional Symmetric Key Cryptosystem is not an efficient scheme to be used for cloud environments.

4.2 Traditional Public Key Cryptosystem Scheme

In this scheme described in [3], the data owner encrypts the file that is desired to be shared with recipients by using every recipient's pubic key so that every one of them is able to decrypt the file with the recipient's own private key. This approach creates a big problem, which is the need of having a different encrypted copy for every recipient. This is very costly in terms of space and computational power.

Accordingly, this scheme does not qualify to be scalable, as it does not provide fine-grained access to the encrypted files and does not support key delegation.

4.3 Broadcast Encryption Scheme

The broadcast encryption scheme in [1] divides all system users into subsets. The data owner in this scheme assigns keys to the users so that every member of a subset S has the same key as the other members of the same subset. The data owner then encrypts the data and broadcasts it to the intended recipients, and they will be able to decrypt it using their own common keys.

This scheme is not efficient as the data owner needs to maintain and refer back to a database for user authorization in order to specify who can access the broadcast channel. Fine-grained access and key delegation are not supported in this scheme, and thus it does not qualify to be scalable.

4.4 Identity Based Encryption Scheme

Every recipient in the Identity Based Encryption Scheme (IBE) pointed out in [3] is assigned a string (could be the user ID) that works as a public key for that recipient. A trusted third party computes a private key derived from that public key. Introducing this third party and delegating the key management to it makes this scheme more scalable. This scheme no longer requires matching the user to his public key, since the user ID is typically the string used as the public key. Key delegation to a trusted third party is achieved in this scheme, but fine-grained access is still a persistent problem.

4.5 Hierarchical Identity Based Encryption Scheme

To take off the high load from the trusted third party in the Identity Based Scheme which performs a very costly task, an implementation has been proposed in [4] to offer two levels of private key generators (PKG): The first level has one Root PKG and the second level has many Domain PKGs. Every domain PKG computes private keys for its corresponding users after it gets its domain secret key from the root PKG, which owns the master secret. Other implementations took this further and offered arbitrary numbers of levels of PKGs. Having more than one level of PGK enhanced the key delegation of the Identity Based Scheme, but didn't provide fine-grained access as well.

4.6 Attribute Based Encryption Scheme

Reference [5] points out a relatively advanced scheme called Attribute Based Encryption scheme (ABE). ABE is considered a generalization of the previously mentioned Identity Based Encryption, in which the user ID is the only attribute used. In ABE, the data owner encrypts a file and specifies a set of attributes for that file and a threshold n. Any recipient needs to have at least n attributes from the file's attribute set to be able to decrypt the file. Other improved implementations support "And" and "OR" logic structures to decrypt the files.

There are two main types of ABE: Key-Policy Attribute-Based Encryption (KP-ABE) where the private key specifies the access structure while the ciphertext is associated with attributes that have to satisfy the access structure in order to enable the user to decrypt the ciphertext, and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) where the private key is associated with a subset of the universe of attributes and the access structure is specified in the ciphertext. The user will only be able to decrypt a ciphertext if the attributes associated with his private key satisfy the access structure specified in the ciphertext. ABE

provides a good level of fine-grained access to data shared by data owner, but does not support key delegation.

4.7 CP-ABE and IBE Hybrid Scheme

The scheme proposed in [9] provides high level of scalability, user privacy, and effective data sharing in the cloud by combining the CP-ABE with IBE. It enables data owners to assign various access privileges for users to the data as well as carry out dynamic requests to adding and revoking access privileges to them. At the same time, the cloud is unable to read any files shared by data owners and saved on the cloud. This scheme is qualified to be scalable as it provides fine-grained access to the encrypted files and supports key delegation.

In this scheme, the data owner specifies an access structure for each file based on a set of meaningful attributes. This access structure can be expressed by an access tree with attributes at leaves and logic gates such as AND and OR as internal nodes. The data owner also assigns an appropriate set of attributes to every user. If the set of attributes for a certain user match with the access structure of the file, the user is granted access to it. Every attribute is assigned a pair of keys: public and private. The private key of an attribute is used along with a user's public key (i.e. user's ID) to generate a secret key component for that user. Combining all the secret key components for a user makes his secret key. This way, we ensure that all system users have different keys. The secret key of the user is used to decrypt the file stored in the cloud if the user's assigned attributes satisfy the access structure of the file. Public key components of the attributes along with the access structure of a file are used to encrypt data files. Figure 1 describes a simplified workflow of the hybrid scheme starting from the system initialization, when the public and private keys of system users are generated and distributed. After that, files are encrypted by the data owner and uploaded to the cloud server. Then, the data owner generates secret keys and delivers them to the corresponding users. At the end, the user will be able to decrypt the files if his attributes match the files attributes and the access structure.

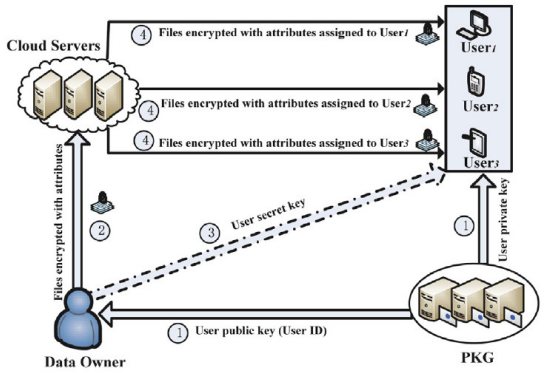

Figure 1. A simplified Workflow of the Hybrid Scheme.

Table I summarizes the criteria that were considered in comparing the techniques stated earlier, including: Fine-grained access, key delegation and anonymity.

Table 1: Comparison between Schemes

Scheme	Fine-Grained Access	Key Delegation	Anonymity
Traditional Symmetric Key cryptosystem	Not Satisfied	Not Satisfied	Satisfied
Traditional Public Key cryptosystem	Not Satisfied	Not Satisfied	Not Satisfied
Broadcast Encryption	Not Satisfied	Not Satisfied	Not Satisfied
Identity Based Encryption	Not Satisfied	Satisfied	Not Satisfied
Hierarchical Identity Based Encryption	Not Satisfied	Satisfied	Not Satisfied
Attribute Based Encryption	Satisfied	Not Satisfied	Not Satisfied
CP-ABE and IBE Hybrid Scheme	Satisfied	Satisfied	Not Satisfied

5 Proposed Solution

The CP-ABE and IBE Hybrid Scheme is considered the most scalable schemes among the previously mention schemes. However, this scheme does not provide data owner anonymity as the identity of data owner is revealed to the cloud during authentication. For example, a hospital patient might be willing to share his health records with his doctor, but doesn't want the cloud to be able to identify that these records belong to this certain patient.

The model presented in [2] introduces a new party to the scene, which is the Security Mediator (SEM). The role of the SEM is to make sure that the data owner is authenticated to the cloud without revealing his identity. This is achieved by delegating signing the file to the SEM instead of the data owner. The SEM can be any typical server from the same organization as the data owner. This model protects the files which the data owner needs to upload, and hides them from the SEM using blinding techniques to enable the SEM to sign the files without knowing their contents.

As shown in Figure 2, the data owner obtains signature on a file he needs to upload to the cloud with the help of the SEM. This is accomplished by first dividing the file into smaller blocks, applying a blinding technique to these blocks, and sending them to the SEM. The SEM in turn signs the blinded blocks using its private key, then sends back the signed blind blocks to the data owner. Upon receiving the singed blocks, the data owner un-blinds them to get the SEM signature on the original block, then the data owner uploads the file singed by the SEM to the cloud server. The cloud server will accept the signature of the SEM since it is part of the same organization. The data user can check the integrity of the uploaded files by sending a challenge that consists of the ID's of the data blocks the data user want to verify along with random numbers to the cloud, and the cloud calculates a response based on this challenge. Based on this response, the data user will be able to determine the integrity of the data blocks.

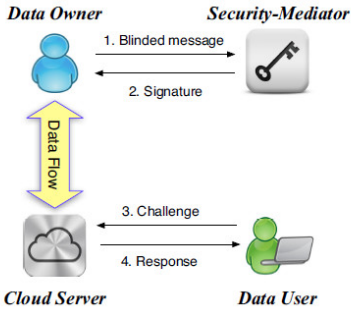

Figure 2. The Security Mediator Model.

Incorporating the highly secure CP-ABE and IBE Hybrid Scheme proposed in [9] with the SEM, we come up with a solution that combines the scalability of the former scheme with the data owner anonymity of the latter model. As shown in Figure 3, the proposed solution is achieved by the following five algorithms:

5.1 Public and private keys generation and distribution

Every data user is assigned a private key and a corresponding public key by a trusted third party called Private Key Generator (PKG). The PKG shares the private key securely with the data user to be able to decrypt the encrypted secret key that will be sent to her by the data owner. The PKG also shares the public key of the same data user with the data owner to be able to use it with the attributes' private keys to create the secret key, encrypt it and send it to the data user.

5.2 Secret key generation and distribution

Every user is assigned a set of attributes by the data owner that need to satisfy the attributes and access structure associated with a file in order to be able to decrypt this file. Every attribute is assigned a pair of keys by the data owner: public and private. The data owner uses the private key of attributes along with the user's public key to generate secret key components for that user. These components combined consist the secret key of the user. Combining the private keys of the attributes with the public key of the user ensures that no users will ever have the same shared key even if they have the same attributes. The data owner encrypts the secret key of the data user with the data user's public key before sending the secret key to her.

5.3 File encryption

The data owner encrypts the file using the public key components of the file attributes along with the access structure of a file as described earlier in the CP-ABE and IBE Hybrid Scheme.

5.4 File signing and uploading

The data owner divides the ciphertext into blocks. Then the data owner applies a blinding technique to the blocks of the ciphertext obtained during the file encryption phase, and send it to the SEM. Applying the blinding technique to the ciphertext prevents the SEM from learning about the content of the data blocks even if the SEM gets hold of any of the authorized users' secret keys. SEM signs the received blocks of ciphertext with its private key, and sends them back to the data owner. The data owner un-blinds the signed blinded block of ciphertext and obtains the original block of ciphertext with the signature of SEM, then uploads them to the cloud. The cloud verifies that the received signature belongs to the SEM. As a

result, the cloud will be able to tell that the user came from a certain organization, but will not be able to identify the data user.

5.5 Checking integrity

To check the integrity of the files uploaded to the cloud, the data user sends a challenge to the cloud specifying the blocks the data user wants to verify along with random numbers. These inputs enable the cloud server to calculate a valid response to the challenge if the file has not been changed. The data user compares the response from the cloud with the values the data user calculates locally based on the random numbers chosen. If the calculated values match at both ends, the data user considers the files to be unmodified.

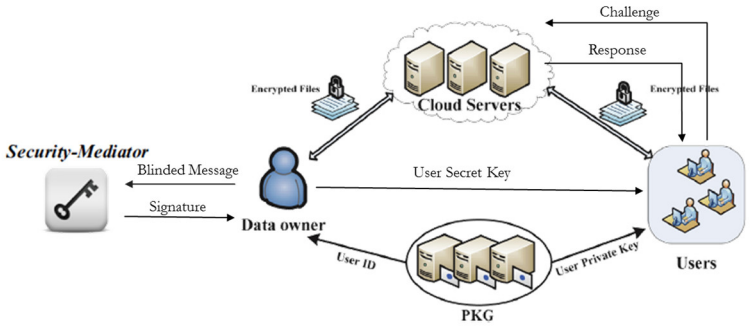

Figure 3. The Proposed Scheme.

6 Advantages of the proposed scheme

The proposed scheme provides a high level of scalability since it supports delegation of key generation and distribution as well as it provides fine-grained access to the data on the cloud. This scheme also preserves the anonymity of the data owner from the cloud server since the cloud is only able to verify that the user belongs to certain organization that is subscribed to the service, but cannot find out the identity of this specific data owner.

The SEM is unable to learn any content from the data it needs to sign after using blinding techniques, which minimizes the requirement of trust on the SEM. The data user can always verify the integrity of the data blocks it needs to verify to be unchanged. This scheme provides less overhead for integrity checking as it enables the data user to choose random blocks of data to verify instead of the entire files, and achieves a very high probability of confidence of the results.

7 Conclusion and Future Work

Many access control schemes have been proposed for managing cloud servers. This paper discussed one of the most secure and scalable schemes which achieves fine grained access control along with key delegation. Since this highly scalable technique does not preserve the data owner anonymity, it is beneficial to integrate this scheme with a model that hides the identity of the data owner by introducing a third party that signs the files on behalf of the data owner. By doing this, the scheme proposed in this paper combines the scalability of the access scheme along with data owner anonymity. It also enables the data user to efficiently verify the integrity of the data uploaded to the cloud.

This paper proposes a new problem for researchers to work on, which is the ability of the scheme not only to support the anonymity of the data owner from the cloud server, but also from the end user. This would

be a great future move especially for researchers. For instance, a patient might allow a medical researcher to view her records stored on the cloud, but does not want to reveal her real identity to the researcher.

REFERENCES

[1] Amos Fiat and Moni Naor. Broadcast encryption. In *Advances in Cryptology CRYPTO93*, pages 480–491. Springer, 1994.

[2] Boyang Wang, Sherman SM Chow, Ming Li, and Hui Li. Storing shared data on the cloud via security-mediator. In *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, pages 124–133. IEEE, 2013.

[3] GuojunWang, Qin Liu, and JieWu. Achieving fine-grained access control for secure data sharing on cloud servers. *Concurrency and Computation: Practice and Experience*, 23(12):1443–1464, 2011.

[4] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *Advances in Cryptology EUROCRYPT 2002*, pages 466–481. Springer, 2002.

[5] Jin-Shu Su, Dan Cao, Xiao-Feng Wang, Yi-Pin Sun, and Qiao-Lin Hu. Attribute based encryption schemes. *Journal of Software*, 22(6):1299–1315, 2011.

[6] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *Parallel and Distributed Systems, IEEE Transactions on*, 24(1):131–143, 2013.

[7] Santanu Chatterjee, Amit Kumar Gupta, and GV Sudhakar. An efficient dynamic fine grained access control scheme for secure data access in cloud networks. In *Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on*, pages 1–8. IEEE, 2015.

[8] Song Lingwei, Yu Fang, Zhang Ru, and Niu Xinxin. Method of secure, scalable, and fine-grained data access control with efficient revocation in untrusted cloud. *The Journal of China Universities of Posts and Telecommunications*, 22(2):38–43, 2015.

[9] Xin Dong, Jiadi Yu, Yuan Luo, Yingying Chen, Guangtao Xue, and Minglu Li. Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing. *Computers & security*, 42:151–164, 2014.