Journal of Universal Computer Science, vol. 21, no. 3 (2015), 454-472
submitted: 30/6/14, accepted: 28/2/15, appeared: 1/3/15 © J.UCS

An Improved Cloud Data Sharing Scheme with Hierarchical Attribute Structure

Zhusong Liu
(School of Computer Science and Technology
Guangdong University of Technology, Guangzhou, China
xiaozhuwuxin@qq.com)

Hongyang Yan
(Department of Computer Science, Guangzhou University, Guangzhou, China
309619240@qq.com)

Zhiqiang Lin
(Institute of Information Engineering, Chinese Academy of Sciences
Beijing, China
linzhiqiang0824@163.com)

Lingling Xu
(Department of Computer Science, South China University of Technology
Guangzhou, China
csllxu@scut.edu.cn)

Abstract: Cloud computing is an emerging computing paradigm that can provide storage resources and computing capacities services over the Internet. However, some new security issues arise when users' sensitive data are outsourced and shared in untrusted cloud. The traditional techniques to protect the confidentiality of sensitive data stored in cloud are encryption and related cryptographic tools. And the corresponding private keys to access and decrypt the files are disclosed to only authorized users. However, these traditional solutions are not scalable because the computational cost of encryption and other access control is heavy for devices with limited computation ability.

In this paper, we present a new way to implement scalable and fine-grained access control systems, which can be applied for big data in untrusted cloud computing environment. The solution is based on symmetric, efficient broadcast encryption and fine-grained attribute-based encryption (ABE). In this access control system, users are able to join and revoked with broadcast encryption. An outsourced Hierarchical ABE scheme is first proposed in this paper to construct the access control system. The security analysis is also presented under the security model.

Key Words: Fine-grained access control, Multi-Authority, Attribute-based encryption

Category: H.2, H.3.7, H.5.4

1 Introduction

With the advent of mobile internet including online social network, a huge number of digital data are generated each day. The users are able to access these information via internet easily. As a result, how to keep the security of sensitive data while allowing efficient access control by authorized users is still a challenge. There are a lot of data access control technologies [Act 1996] to effectively implement fine-grained access control. They are able to provide not only flexibility but also differential access rights for users. However, traditional methods for the design of access control rely on the assumption that data owners and storage servers are in the same trusted domain. That is to say, the users believe and trust the storage servers will behave honest and perform the data according to the requests from the users. As a new and novel computing paradigm, cloud computing attracts much attention recently from both academia and industry. The techniques of virtulizations, utility computing, Service Oriented Architecture have been used in the cloud computing, which provides unlimited resources to users on-demand. However, the users do not need to get the knowledge of the platform and implementation details for the cloud computing, which has more advantages over the previous computing paradigms. Furthermore, With this paradigm, the users are providing both computation and storage services from the cloud server, which can help user perform intensive tasks for users with limited computational ability by outsourcing to cloud servers. As a result, the users greatly reduce the local cost for computation and storage with the aid of cloud computing. Of course, with the utilization of cloud computing, the users are also able to get many other services such as high reliability and scalability for the storage or computing services from the cloud system. Nowadays, there existed many popular commercial cloud computing platforms such as Amazon's EC2 and S3 [Cloud AEC 2011], Google App Engine [Zahariev, A. 2009], and Microsoft Azure [Vecchiola et al. 2009].

Obviously, such an assumption is too strong and cannot be used in cloud computing because the entities of the data owners and cloud storage servers do not belong to the same trust domain. Furthermore, we also need to prevent the cloud server from access the users' data because the cloud server is also not trusted. Usually, there are some security techniques which could guarantee the security of outsourced data, such as the traditional encryption methods. The sensitive data will be encrypted before they are outsourced to the cloud. However, traditional data encryption is not efficient and the utilization of data is becoming a challenge in Cloud Computing. With the huge number of data to be encrypted and shared by users with mobile devices, such kind of sharing operation with privacy is difficult to achieve. Therefore, some new efficient sharing methods should be provided for these data outsourcing.

With the attractive properties of cloud computing, more and more personal data have been moved by the individuals and organizations from the local computers to the cloud computing platform. The ever-increasing amount of valuable digital data both at home and in business can be processed by the cloud computing. Furthermore, the users can also share these data with eligible users through cloud computing techniques in a secure way. Cloud computing is able to offer user-friendly, easily accessible ways to store and share data between users and synchronization of multiple devices. Actually, with the development of online mobile social network, more and more sensitive personal data, such as private photos, videos and sensitive documents, is shared and stored by third-party sites on the Internet. Therefore, when the data are moved by users from local storage to cloud computing, the security of them should be taken into accounted because the cloud server is public and untrusted by users. Therefore, to share the data through cloud computing, users have utilize secure access control system which operated in an open environment.

As prevention of unauthorized access can be generally achieved by encrypting sensitive contents before uploading them to cloud servers, there still exist many challenge issues for its implementation in practical systems. In particular, differentiated content access is frequently required in the sense that users with different roles (or paying different prices) should be granted different level of access privileges. For the purpose of helping the data owner impose fine-grained access control of data stored on untrusted cloud servers, a feasible solution would be encrypting data through certain cryptographic primitive(s). The data owner only discloses decryption keys to authorized users. This general method actually has been widely adopted by existing works which aim at security of data stored on untrusted servers. One critical issue of it is how to achieve the security goals without having high computational cost at key management. Existing work resolve this issue either by introducing a per file access control list (ACL) for fine-grained access control, or by categorizing files into several file groups for efficiency. When the system scales, such a method will introduce heavy computation burden at the user side and thus, it is only able to provide coarse-grained access control of data.

To support fine-grained and efficient access control, attribute-based encryption (ABE) [Sahai and Waters 2005] has attracted much attention in the research community to design flexible and scalable access control systems. One of the most attractive properties of ABE is that it enables public key based one-to-many encryption. Thus, many systems have used this kind of encryption to realize scalable and fine-grained access control systems, where different yet flexible access rights can be assigned to individual users. There are two kind of ABE proposed to address complex and general access policy [Goyal et al. 2006], that is, key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In KP-ABE, access policy is assigned in attribute private key, whereas, in CP-ABE, the access policy is specified in the ciphertext.

Since its inaugural, ABE is envisioned as a highly promising public key primitive for realizing scalable and flexible access control systems as for the first time ABE enables public key based one-to-many encryption. It assigns differential access rights to a set of users and allow flexibility in specifying the access rights of individual users. A user has an access only if there is a match between the attributes of the ciphertext and the user's key. Although ABE provides secure and fine-grained access control, before its deployment in cloud computing, the efficiency issue should be considered. For scalability purpose, it is necessary to enable efficient encryption and decryption operations.

1.1 Contributions

The contributions of this paper in addressing the secure fine-grained access control problem in Cloud Computing is multi-fold.

Firstly, for each file, we achieve to define and enforce access policies based on attributes in the system. To improve the efficiency of access control based on traditional ABE, we utilize the notion of hierarchical ABE in the system. Furthermore, we show how to outsource the decryption time at the user side by proposing an outsourced hierarchical ABE construction. In this way, through reducing the encryption and decryption time at user side, the efficiency for the system can be greatly improved. Furthermore, we also show the security proof for the system.

1.2 Related Works

1.3 Attribute-based Encryption

As a new one-to-many public encryption, there are many research results in the topic of attribute-based encryption. To improve the encryption efficiency, attribute based encryption [Sahai and Waters 2005] has been proposed and applied in the access control in cloud computing. Then, a lot of other schemes supporting more fine-grained access control were also proposed such as [Chase 2007, Goyal et al. 2006, Bethencourt et al. 2007, Li et al. 2010]. Attribute-based encryption (ABE) is able to realize fine-grained access control in public cloud systems because the cloud servers are not able to get the messages from the ciphertext and the efficiency is better. In traditional public key encryption, the public parameters and ciphertext grows linear with the number of users in the system. However, in attribute-based encryption, such a limitation is overcome based on its special property. That is to say, the public parameters and ciphertext is only related with the number of attributes, instead of the number of users. In this way, differential privileges and fine-grained access rights can be issued to users in the system. In ABE, according to the policy definition, two different primitives are defined, that is, the key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE) [Goyal et al. 2006]. In the definition for KP-ABE, when the user requests the private key, the attribute authority will issue corresponding policy embedded in the private key such that the user cannot change its policy in the private key. The private key issued to the user can be used by the user to download the encrypted files and decrypt them. Different from KP-ABE, in the definition of CP-ABE system, the attribute authority only issues private keys according to the attributes for the users. The access policy or structure will not be considered in the key issue phase by the attribute authority. However, the generation of the ciphertext will take this factor into account and specified in each file. Thus, the access policy is controlled by the data owner, not the attribute authority. In this way, based on these two primitives, different systems can adopt corresponding policy in terms of their requirements in the applications. We explore how to utilize CP-ABE to manage the files stored in cloud computing. Baek *et al.* showed how to shorten the public parameters of [Sahai and Waters 2005], but the scheme could only be proved to be secure in the random oracle model. Another important application of attribute-based encryption is access control [Sahai and Waters 2005]. Goyal *et al.* [Goyal et al. 2006] proposed key-policy attribute-based encryption. The access structure in KP-ABE is embedded in the private keys and cannot be changed or determined in the ciphertext.

However, all the previous ABE works only considered the single attribute-authority setting. The security of these systems relies on the assumption that the central attribute authority is trusted. To solve this challenge, Chase [Chase 2007] proposed a multi-authority attribute-based encryption scheme to reduce the trust of attribute authority. They introduced multiple authorities to solve the trust assumption. That is, in this multiple authority protocol, each authority controls some of the attributes. To implement fine-grained access control, they [Chase 2007, Wu et al. 2012] also extend their construction in tree-structure as [Goyal et al. 2006]. Another related notion with ABE is hidden identity-based signature. It also achieves similar function with ABE to realize the authentication. The notion can be viewed as the combination of ABE and group signature. In such a primitive, the eligible user can generate a valid signature with their attribute private key. Upon receiving the signature, any user is able to confirm if such a signature is from some user with the attributes included in the signature. It has some difference with the group signature because the user who generates the signature is kept hidden even to the authority, that is, it can achieve unconditional anonymity.

ABE can be viewed as the extension of identity-based encryption (IBE), which was firstly implemented by Boneh and Franklin [Boneh and Franklin 2001].

Nevertheless, it has the limitation for scalable system, especially for the system supporting efficient user revocation. To revoke the users in IBE, Hanaoka et al. [Hanaoka et al. 2005] proposed a trivial method to periodically renew their private keys without interacting with PKG. However, the assumption required is that each user has to keeps a tamper-resistant device, which makes their system is impractical and ineffective. Another recent solution is the introduction of mediator-aided revocation. There is a mediator in the system who can help to decrypt each ciphertext. Then, for any revoked user belonging to the revocation list, the mediator will not help him decrypt the ciphertext anymore. In this way, the revocation is realized. However, such a third party assumption is too strong and cannot be implemented in many applications. The notion of proxy re-encryption is also a very useful notion proposed to realize a revocable ABE scheme. In this primitive, the cloud server can change the receivers by re-encrypt the files. There is another trusted authority who is able to update master key according to attribute revocation status in each time period and issue proxy re-encryption key to cloud. The cloud server will then re-encrypt ciphertext using the re-encryption key. The revoked users cannot access and decrypt to get the files anymore without correct private keys.

1.4 Cloud Computing

Cloud Computing is the latest term providing computing resources and storage resources as a service [Yakut and Polat 2012] [Choy et al. 2005] [Lin and Lo 2013] [Chen et al. 2012] [Wu et al. 2009] [Chu et al. 2008]. In many applications, the storage can be provided to the users as a service of infrastructure (IaaS) model. Thus, this notion provides the many tools to get materials of cloud computing. Such a notion is different from the traditional hosting services because the clouds are available to users with internet anytime and anywhere, which is rented by the users on a monthly or yearly basis in this environment. More specifically, the cloud infrastructure is rented as virtual machines on a per-use basis and is dynamically scale according to the users' requests. One of the advantages is that all the users do not need to know the detailed information about the cloud infrastructure or the details of operation in cloud computing. Nowadays, with the developments of cloud computing, there are more and more cloud storage applications such as EC2 and S3, which can provide different storage services. In our paper, we assume that there is such kind of outsourcing storage in practice and could be utilized.

Another important advantage of cloud computing is that the users can be provided the computing services from it, that is, the user can utilize the cloud computing and outsource their expensive computations to the cloud servers. In literature, there are also many works proposed to solve the secure outsourcing computation problem in cloud computing, that is, the computing as a service. In

2011, Green et al. [Green et al. 2011] proposed a new method for efficiently and securely outsourcing ABE, which can relieve the users' computation overhead. In their paper, the cloud server is assumed to act honestly and try to get much information from the resources. In this kind of outsourcing computation, the users need to transform their private keys before outsourcing. There are a lot of key blinding methods such as multiplication with factor or addition of some random number. As a result, the user can keep their sensitive information while relieving their computation overhead. There are also some other works about outsourced ABE include [Li et al. 2013, Li et al. 2013, Li et al. 2014, Chen et al. 2013, Chen et al. 2013].

1.5 Organization

Some preliminaries on bilinear pairings and assumptions are given in the next Section. In Section 3, the definitions and models for access control problem in cloud computing are presented. Furthermore, an HABE scheme and the construction of fine-grained access control is described in Section 4. Its security analysis and performance analysis are given in this Section 5 and Section 6, respectively. This paper ends with some concluding remarks.

2 Preliminaries

We now give a brief revision on the property of pairings and some candidate hard problems from pairings that will be used later.

2.1 Bilinear Pairings

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic groups of prime order p, writing the group action multiplicatively. Let g be a generator of $\mathbb{G}_1$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a map with the following properties:

1. Bilinearity: $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$ for all $g_1, g_2 \in \mathbb{G}_1$, and $a, b \in_R \mathbb{Z}_p$;
2. Non-degeneracy: There exists $g_1, g_2 \in \mathbb{G}_1$ such that $\hat{e}(g_1, g_2) \neq 1$, in other words, the map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$;
3. Computability: There is an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}_1$.

Two assumptions are also used in our paper, *i.e.*, CDH assumption and DBDH assumption, which our proof is based on.

CDH Assumption. The Computational Diffie-Hellman assumption is that, given

g, g^x, $g^y \in \mathbb{G}_1$ for unknown random $x, y \in \mathbb{Z}_p^*$, it is hard to compute g^{xy}.

DBDH Assumption. The Decision Bilinear Diffie-Hellman assumption is that, given g, g^x, g^y, $g^z \in \mathbb{G}_1$ for unknown random $x, y, z \in \mathbb{Z}_p^*$, $T \in \mathbb{G}_2$, it is hard to decide if $T = \hat{e}(g, g)^{xyz}$.

2.2 Symmetric encryption

Symmetric encryption uses a common secret key κ to encrypt and decrypt information. A symmetric encryption scheme consists of three primitive functions:

- $\mathsf{KeyGen}(1^\lambda) \rightarrow \kappa$ is the key generation algorithm that generates κ using security parameter 1^λ;
- $\mathsf{Encrypt}(\kappa, M) \rightarrow C$ is the symmetric encryption algorithm that takes the secret κ and message M and then outputs the ciphertext C; and
- $\mathsf{Decrypt}(\kappa, C) \rightarrow M$ is the symmetric decryption algorithm that takes the secret κ and ciphertext C and then outputs the original message M.

2.3 Broadcast Encryption

A broadcast encryption scheme is demanded in this paper. There are four algorithms defined in a broadcast encryption scheme, $KeyGen_{BE}$, Enc_{BE}, Dec_{BE}, and Rev_{BE}. Note that in the broadcast encryption scheme, it also provides revocation function to protect the security against a coalition of all revoked users [Chor et al.]. The description of the algorithms is as follows,

- $KeyGen_{BE}$ is the key generation algorithm that is used to generate a long-lived key for the user.
- Enc_{BE} is the encryption algorithm that is used to encrypt files to a privileged user group G.
- Dec_{BE} is the decryption algorithm that is used to decrypt the ciphertext by authorized users. Assume that a message is encrypted to a user group G. Then it means that only users in the group G can decrypt and get the message from the ciphertext.
- Rev_{BE} is the revocation algorithm that is used to revoke the users from the broadcast user group G. Any user can be removed from the valid and privileged user group G. In this way, it means that the revoked user cannot decrypt the ciphertext anymore.

3 Models and Basic Techniques

3.1 System Model and Security Model

There are two entities in our system. That is, we consider a cloud data system consisting of data users and cloud servers. Note that cloud storage is a model of networked enterprise storage where data is stored in virtualized pools of storage which are generally hosted by third parties.

– Data users. The users in the system are issued with the privileges to access cloud data. They can use the private keys to decrypt the data downloaded from the cloud storage servers. The user will send request of data retrieving to the cloud servers after data outsourcing. The request could be from the data owner or from the users with the privileges to get the files. In this paper, we assume that the cloud servers do not need to check the privileges of the users before sending the encrypted files to the users, which is based on our trust assumption.

– Cloud Servers. Cloud Servers are always online and operated by Cloud Service Provider (CSP). They are assumed to have abundant storage capacity and computation power. The cloud servers in our paper only need to provide the storage service to the users without any other computing service. Furthermore, the cloud servers are assumed that they will honestly answer the requests sent from the users such as the download of files. Apart from this, the cloud servers are also assumed that they will store the users' data correctly with integrity. Thus, the users' data will not be lost or damaged in our system.

In this work, we consider Honest but Curious Cloud Servers. That is to say, Cloud Servers will follow the proposed protocol, but try to find out as much secret information as possible based on their inputs. Such an assumption is reasonable and popular model in the current cloud computing research. In more details, we assume that the cloud servers will not damage the users' data or answer the requests from the users in an incorrect way. What the cloud servers are interesting is the data itself, that is, the cloud server would like to get the useful information of the encrypted data. In this paper, the communication between the users and the cloud servers need not secure communication channels because all the information sent from or to users are encrypted. However, the communication channel between the users and the attribute authority is assumed to be secure. The users need to get the private key from the attribute authorities, which should be protected and prevented from the attackers. Of course, if each user owns some public key, such a secure channel is also not necessary because the attribute private keys could be encrypted and sent to the users. Users are able to get the access to their data with corresponding correct privileges keys. They could download the encrypted files and decrypted the data by using the decryption algorithm such as the attribute-based encryption in our constructions. Obviously, if the users do not have the right privileges, they are unable to decrypt the encrypted files with the incorrect private keys. This security can be guaranteed from the security assumption of attribute based encryption.

4 Access Control System For Data Sharing

4.1 Basic Tool: HABE

In this section, we give the introduction of HABE with threshold key policy, including the definition, security definition and construction in [Li et al. 2011]. In HABE, when one encrypts a message m for a set of target attributes $\{\omega_1, \cdots, \omega_k\}$, anyone can decrypt the ciphertext if his attributes matches the policy in the ciphertext, that is, the user is able to decrypt the ciphertext if he has at least d attributes that cover the target attributes $\{\omega_1, \cdots, \omega_k\}$.

There four algorithms (Setup, KeyGen, Enc, Dec), defined in the HABE system, which can be described as follows:

- Setup: The setup algorithm is a probabilistic polynomial time algorithm, which takes input the security parameter 1^λ, it generates public parameters para for the system and the master private key sk. It sends the master key sk to the attribute center via a secure channel and outputs para as public parameter, including attributes set and attributes structure.
- KeyGen(U,para, sk): The private key generation algorithm is a probabilistic polynomial time algorithm, which takes input attribute subset U and master key sk. The algorithm outputs a private key d_{U} with sk for a user with attribute set U.
- Enc(m,U$'$,para): The encryption is a probabilistic polynomial time algorithm, which takes input a message m and public parameters para, and attribute set U$'$ for the intended receivers. Finally, it outputs the ciphertext $\mathcal{C}$.
- Dec($\mathcal{C}$,U$'$,U,d_{U}): The decryption algorithm is a deterministic algorithm, which takes input a ciphertext $\mathcal{C}$ for U$'$, and public parameters para, and secret key d_{U} with respect to U. It checks if the number of elements in U that cover U$'$ is at least d. If it is, outputs the plaintext m with d_{U}. Otherwise, outputs $\perp$.

The security requirements for HABE is indistinguishable against adaptively chosen attributes and chosen ciphertext attacks (IND-Atr-CCA).

The formal definition for IND-Atr-CCA is based on the following game involving an adversary $\mathcal{A}$.

Game IND-Atr-CCA

- **Setup**(d). The challenger chooses a sufficiently large security parameter 1^λ and runs Setup to get key pair (pk, sk) and other public parameters para. Retain secret key sk and gives pk, para to $\mathcal{A}$.
- **Phase 1**. $\mathcal{A}$ can perform a polynomially bounded number of queries to the oracles in an adaptive manner, including attributes private key extraction oracle and ciphertext decryption oracle.
- **Challenge**. $\mathcal{A}$ outputs an attribute identity U^* and two messages m_0, m_1 on which it wishes to be challenged. The only restriction is that $\mathcal{A}$ did not previously issue a key query on U such that the number of elements in U that cover some element in U^* is not less than d. The challenger randomly chooses a bit $b \in \{0, 1\}$, computes $\mathcal{C} = \mathsf{Enc}(m_b, \mathsf{U}^*, \mathsf{para})$ and sends $\mathcal{C}$ to $\mathcal{A}$.
- **Phase 2**. $\mathcal{A}$ can perform a polynomially bounded number of queries to the decryption and private key extraction oracles in an adaptive manner. $\mathcal{A}$ is not allowed to issue decryption query on $(\mathcal{C}, \mathsf{U})$ or private key query on U for any attributes U such that the number of elements in U that cover some element in U^* is at least d.
- **Guess**. $\mathcal{A}$ outputs a guess bit b'.

$\mathcal{A}$ wins the game if $b = b'$. The advantage of $\mathcal{A}$ in game $\mathsf{IND\text{-}Atr\text{-}CCA}$ is defined as the probability that $\mathcal{A}$ wins the game minus $1/2$.

There is also another weaker notion called indistinguishable against selective attributes and chosen plaintext attacks ($\mathsf{IND\text{-}sAtr\text{-}CPA}$). The definition is the same with $\mathsf{IND\text{-}Atr\text{-}CCA}$, except here it requires the adversary submits its challenge attributes U^* before the setup phase.

4.2 An Improved HABE Construction with Outsourced Decryption

Though the notion of HABE improves the efficiency of ABE in many applications with hierarchical structure, the decryption cost at the user side is still very high. In this section, we show how to present an new HABE construction with outsourced decryption, which can be applied in many application with mobile devices. Two more algorithms are defined in the outsourced HABE scheme, including the outsourcing key generation algorithm (OutKeyGen) and server-aided decryption (OutDec). The constructions are described as follows.

Setup(d): The bilinear pairing parameters are chosen. That is, let the group of G_1 be the bilinear group with prime order p. The element of g is its generator. $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is a bilinear map over the groups given. In this construction, we define N attributes in universe. We also assume that there are n trees are formed in the access control system. A hash function $H : \{0,1\}^* \rightarrow Z_p^*$ is defined. Let U_0=$\{\omega_{10}, \cdots, \omega_{n0}\}$ be the root attribute set. Assume the maximum depth of the i-th tree is ℓ_i for $1 \leq i \leq n$, and $\ell = \max\{\ell_1, \cdots, \ell_n\}$. The value of α from Z_p is chosen as a secret and the user computes $g_1 = g^{\alpha}$. The user also randomly chooses g_2, $u'_1, \cdots, u'_n$, $u_1, \cdots, u_\ell$ from group G_1.

The user publishes the system parameter $\mathsf{para} = (g_1, g_2, \hat{e}, (u'_i)_{1\leq i\leq n}, (u_i)_{1\leq i\leq \ell})$. The secret key α is kept secret.

KeyGen: Upon receiving a request of key issuing from a user, the authority proceeds as follows:

- A $d-1$ degree polynomial q is randomly chosen such that $q(0) = \alpha$;
- For each $\omega \in \mathsf{U}$, assume its depth is k in the i-th tree with path $(\omega_{i0}, \omega_{i1}, \cdots, \omega_{i,k-1}, \omega)$. It chooses $r \in_R Z_p$ and computes D_ω=$(d_{i0}, d_i, d_{i,k_i+1}, \cdots, d_{i\ell_i})$, where $d_{i0} = g_2^{q(H(\omega))}(u'_i \Pi_{j=1}^{k} u_j^{\omega_{ij}})^r$, $d_i = g^r$, $d_{i,k+1}$=u_{k+1}^r, $\cdots$, $d_{i\ell_i}$=$u_{\ell_i}^r$;
- Finally, it outputs the private key of U as

$$d_{\mathsf{U}} = \{D_\omega\}_{\omega\in\mathsf{U}}$$

Enc: To encrypt a message $m \in G_2$ to an attribute set U', it proceeds as follows. First, a random value $s \in Z_p$ is chosen. For each $\omega' \in \mathsf{U}'$, assume its depth is k' in the j-th tree. Let the path for ω' be $(\omega_{j0}, \omega'_{j1}, \cdots, \omega'_{j,k'-1}, \omega')$. It computes $E' = m\hat{e}(g_1, g_2)^s$ and $T = g^s$. Furthermore, it computes $E_{\omega'} = (u'_j \prod_{\delta=1}^{k'} u_\delta^{\omega'_{j\delta}})^s$ for each $\omega' \in \mathsf{U}'$ and outputs the ciphertext as

$$\mathcal{C} = (E', T, \{E_{\omega'}\})$$

for all $\omega' \in \mathsf{U}'$.

OutKeyGen: Suppose that a user has the private key $d_{\mathsf{U}} = \{D_\omega\}_{\omega\in\mathsf{U}}$ and he wants to outsource the key to the cloud server for decryption. He just computes $d'_{\mathsf{U}} = \{D_\omega^s\}_{\omega\in\mathsf{U}}$ by choosing a random $s \in Z_p$. Then, the outsourced key is sent to the cloud server.

OutDec: Suppose that a ciphertext E is encrypted to the attribute set U'. Assume one has a private key $d_{\mathsf{U}} = \{D_\omega\}_{\omega\in\mathsf{U}}$ for attribute set U such that the number of attributes in U that cover the attributes in U' is no less than d. Then, it chooses an arbitrary d-element subset S with elements in U. For each ω in S with path $(\omega_{i0}, \omega_{i1}, \cdots, \omega_{i,k-1}, \omega)$, assume ω' is the attribute in U' covered by ω with path from the same root ω_{i0} as $(\omega_{i0}, \omega'_{i1}, \cdots, \omega'_{i,k'-1}, \omega')$ (It implies the depth for ω' is k' in the j-th tree). Then, we have $\omega_{i\delta} = \omega'_{i\delta}$ for $1 \leq \delta \leq k$. These public values are sent to the cloud server for the decryption. Upon receiving the decryption request, the cloud server computes $d'_{i0} = d^s_{i0} d^{s\omega'_{i,k+1}}_{i,k+1} \cdots d^{s\omega'_{ik'}}_{ik'}$ and decrypts the ciphertext partially as $C' = \prod_{\omega \in S} (\frac{\hat{e}(d'_{i0}, T)}{\hat{e}(d^s_i, E_{\omega'})})^{\triangle_{H(\omega),S}(0)}$.

Dec: After receiving the partially decrypted ciphertext C', the user decrypts the ciphertext with his private key as

$$m = E' / C'^{\frac{1}{s}}$$

4.3 Access Control System based on Hierarchical Attribute-based Encryption

We first propose a construction by using the broadcast encryption technique. For efficiency, hybrid encryption method is applied here. More specially, the data files are encrypted with symmetric keys, which are encapsulated with the broadcast encryption.

- To upload a file F, a key K is chosen as the key for SE and the file is encrypted by computing $C = Enc_{SE}(K, F)$. After this, a user set U_i is defined and the key is encrypted with these users' key as $C = Enc_{BE}(U_i, K)$.
- Assume a user in U_i has been issued a long-lived secret key sk_i of the broadcast encryption scheme. This long-lived key is issued by the data owner at the setup of the system to each authorized user, including the cloud servers. This user can decrypt and access the file by computing $K = Dec_{BE}(sk_i, C)$ with sk_i. After decrypting to get K, he can decrypt to retrieve the file as $F = Dec_{SE}(K, C)$.

If there is a user wants to join the system, he will be issued a private key from the authority for the user. The grant algorithm is same as the corresponding algorithm in the broadcast encryption algorithm. Similarly, the revocation can be also implemented with the corresponding revocation algorithm in broadcast encryption BE. Though the above construction can provide the access control for untrusted cloud server, its efficiency is not so good for scalable system.

4.4 Access Control System based on Hierarchical Attribute-based Encryption

We show how to construct a scalable access control system with HABE technique in cloud computing.

- To upload and share an file with specific privileges, that is, to upload an file which can only be accessed by users with specific users, the data owner determines the access policy of W. A secure secret key K is chosen too. He encrypts F with the symmetric encryption as $C = Enc_{SE}(K, F)$. Similar to the above construction, such a symmetric encryption key can be used to encrypt the data with same privileges. Of course, the data owner can also select a key for each file such that the file or policy can be dynamic with better efficiency. After computing these values, the secret key K is encapsulated with the encryption algorithm of attribute-based encryption under access policy W.

- Each user will be issued an attribute private key according to his privileges. The data owner defines and determine the access policy for each user in this system for each file. That is, he generates a privilege key by using the key generation algorithm in the ABE scheme.

- After receiving the secret key for the access policy, the user can check and find if his attributes match the access policy given for the file. If it matches, the user can get the file F by computing the decryption algorithm of $Dec_{SE}(K, C)$.

Because the access policy for each file is defined through attributes, this system enables fine-grained access control based on the application of attribute-based encryption. However, it has efficiency bottleneck and will be very inefficient to realize the user revocation with the increase of scalability: Whenever there is a user to be revoked, the traditional method is to update all the attribute private keys for the authorized users [Sahai and Waters 2005].

4.5 System Description

Setup In our construction, we assume that the universe of attributes in the system is denoted by $U = \{\omega_1, \omega_2, \cdots, \omega_n\}$. For a given security parameter λ, this algorithm obtain the public parameter $para$ and the master key sk. The public parameter includes the public key for the attribute-based encryption, broadcast encryption schemes etc. The master key for the attribute authority will be sent to the attribute authority through secure channel.

New File Creation Whenever the data owner creates and uploads a file F to the cloud servers, he first defines an access policy W for this file. A secret key K for the symmetric encryption is chosen. Then, with the given symmetric encryption algorithm, F is encrypted with K. A ciphertext C is obtained by using the encryption of K for policy W. The encrypted file and the encrypted key are uploaded to the cloud servers.

New User Join Assuming a new user wants to join the system, a private key for his privileges will be issued from the attribute authority. In concrete, the attribute authority assigns a set of attributes L by running the key generation algorithm of attribute-based encryption.

File Access Suppose a user wants to access and retrieve files of his interest. The following steps will be taken. Firstly, the cloud servers send the requested encrypted files to the user. If the user has the privilege to access the encrypted file, he can first outsources the decryption task by computing a outsourced private key to the cloud server and get a partially decrypted ciphertext. Then, he can fully decrypt by using the attribute-based encryption to get the symmetric key K with the *Decrypt* algorithm. The decryption in the symmetric encryption $Dec_{SE}(K, C)$ will be proceed on the encrypted data C and the file F can be retrieved.

User Revocation Suppose that a user with identity ID' need to be revoked. The data owner needs to guarantee that the revoked user cannot get the key for the access to the files which the revoked users are not allowed to access. To achieve this, the data owner utilizes the broadcast encryption to encrypt another random number r' for a group of users without the revoked user. The previous ciphertext before revocation will be deleted by the cloud server and data owner. In this way, the user revocation can be realized.

4.6 Attribute-based Encryption Supporting Multiple Authorities

In our above construction, there is only one attribute authority, which has to be fully trusted by the users in the system. To reduce the trust on the attribute authority, we consider how to construct another improved system with multiple authorities. In an attribute-base encryption system with only one authority, the user must go to the attribute authority, and prove his identity in order to obtain an attribute private key which will allow him to access the cloud files. As a result, all the users have to ask the attribute private keys from the authority with proof that he has a certain set of attributes. Then, the corresponding private keys will be issued from the authority. However, this means we must have one trusted server to monitor all attributes, such as driver's licenses and education degree. In reality, we have two different entities responsible for maintaining this information, so we would want to be able to entrust each of these to different authorities. To tackle this problem, the notion of multi-authority ABE [Chase 2007] will be used where each attribute-authority issues private keys to user according to respective domain of attribute.

In the multi-authority ABE [Chase 2007], a novel construction technique has been used to support collusion attack from the multiple authorities. Supporting multiple authorities in ABE requires more than the standard techniques in distributing cryptosystems due to the collusion resistance requirement. In attribute-based encryption, one of the most important security notion is to prevent collusion of users with different access privileges. In another word, even they collude together, the users are not allowed to get files that are not belonging to their privileges. In more details, the private key is unique for each user and cannot combined directly by different users. The security requirement asks the attribute authority to issue the private keys without knowing each other's private keys and they cannot be shared by different authorities. Chase and Chow [Chase and Chow 2009] proposed a multi-authority ABE with an anonymous key issuing protocol, where a novel pesudo-random function is utilized such that the sum of these values will be zero and canceled in the algorithm of decryption.

5 Security Analysis

As described in the system security model, three kinds of external adversaries have been considered in our system, that is, the public cloud servers, the revoked users and other users without corresponding attributes. In another word, we want to show the security of our scheme that all the external attackers are not able to learn any partial information about the underlying files. First, we are able to prove for revoked users, they cannot get the random key encrypted by the broadcast encryption, which is only for eligible users. Based on the security of broadcast encryption, we can easily get the security of our construction against revoked users. That is, they cannot get the valid key and data from the ciphertext. Then, we analyze the security for the public cloud servers, remember that the public cloud servers are not provided any private attribute private keys. This means that the attack ability is weaker than the illegal users, with similar reason, these cloud servers are not able to retrieve any valid information from the ciphertext. Therefore, we can prove that our construction is also secure against the public cloud server based on the security of broadcast encryption. The strongest attackers are the users which are not the valid receivers for the ciphertext. For these users, they have valid attribute private keys and access to the ciphertext stored on the public cloud servers. Thus, they are able to decrypt the ciphertext for the broadcast encryption if they are included in the list. However, we have another second level for the security, that is, the attribute-based encryption. If the attacker's attributes do not match the access policy, they cannot get the underlying secret key used in the symmetric encryption. In another word, the attackers are not able to decrypt and retrieve the file. Furthermore, our attribute-based encryption is secure against collusion attack, this means that even if the attackers collude, they are unable to get the information from the ciphertext either. The outsourcing technique we used in this protocol is from [Green et al. 2011]. Thus, from the security of this outsourcing technique, we know that the cloud server still cannot get any useful information from the ciphertext and the outourced private keys.

From the security analysis we know that our system can be proved to be secure if the HABE construction is secure. The given construction has been proven in the security against chosen message attack.

6 Conclusion

This paper addresses the problem of fine-grained access control system in cloud computing. We demonstrate the challenges of implementing fine-grained access control in cloud computing by using the existed technologies to tackle the issue. We propose an ABE construction with hierarchy structure to implement access control in cloud computing. To reduce the computational overhead at the user side, the construction of outousrced HABE scheme is also proposed. The security analysis also demonstrates that the construction is secure against the adversary with the chosen message attack and collusion attack, that is, it is secure with respect to the data confidentiality and fine-grained access control. We also give the performance analysis and show that the construction is efficient.

Acknowledgement

This work was supported by the cooperation project in industry, education and research of Guangdong province and Ministry of Education of China under Grant No.2012B091000073, National Natural Science Foundation of China (No. 61472091), Natural Science Foundation of Guangdong Province (Grant No. S2013010013671), and the Guangzhou Zhujiang Science and Technology Future Fellow Fund (Grant No. 2012J2200094).

References

[Abdalla et al. 2006] Abdalla, M., Catalano, D., Dent, A. W., Malone-Lee, J., Neven, G., Smart, N. P.: "Identity-based encryption gone wild"; Automata, Languages and Programming. Springer Berlin Heidelberg, 2006. 300-311.

[Act 1996] Act, A.: "Health insurance portability and accountability act of 1996"; Public Law 104 (1996): 191.

[Au et al. 2008] Au, M. H., Huang, Q., Liu, J. K., Susilo, W., Wong, D. S., Yang, G.: "Traceable and retrievable identity-based encryption"; Applied Cryptography and Network Security. Springer Berlin Heidelberg, 2008.

[Bethencourt et al. 2007] Bethencourt, J., Sahai A., Waters B.: "Ciphertext-policy attribute-based encryption"; Security and Privacy, 2007. SP'07. IEEE Symposium on. IEEE, 2007.

[Boneh and Franklin 2001] Boneh, D., Franklin, M.: "Identity-based encryption from the Weil pairing"; Advances in Cryptology-CRYPTO 2001. Springer Berlin Heidelberg, 2001.

[Chen et al. 2012] Chen, X., Li, J., Ma, J., Tang, Q., Lou, W.: "New algorithms for secure outsourcing of modular exponentiations"; Computer Security-ESORICS 2012. Springer Berlin Heidelberg, 2012. 541-556.

[Cheung and Newport 2007] Cheung, L., Newport, C.: "Provably secure ciphertext policy ABE"; Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007.

[Chase 2007] Chase, M.: "Multi-authority attribute based encryption"; Theory of Cryptography. Springer Berlin Heidelberg, 2007. 515-534.

[Chase and Chow 2009] Chase, M., Chow, S. S.: "Improving privacy and security in multi-authority attribute-based encryption"; Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009.

[Chen et al. 2012] Chen, X., Li, J., Susilo, W. : "Efficient Fair Conditional Payments for Outsourcing Computations". IEEE Transactions on Information and Forensics Security (TIFS), 7(6): 1687-1694, 2012.

[Chen et al. 2013] Chen, X., Li, J., Ma, J., Wong, D., Lou, W.: "New and Efficient Conditional E-payment Systems with Transferability". Future Generation Computer Systems, Elsevier, 2013.

[Chen et al. 2013] Chen, X., Li, J., Ma, J., Tang, Q., Lou, W.: "New Algorithms of Outsourcing Modular Exponentiations". IEEE Transactions on Parallel and Distributed Systems, 2013.

[Chen et al. 2013] Chen, X., Li, J., Huang, X., Li, J., Xiang, Y.: "Secure Outsourced Attribute-based Signatures". IEEE Transactions on Parallel and Distributed Systems, 2013. http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.2295809

[Chu et al. 2008] Chu, C., Tzeng, W.: "Efficient k-out-of-n Oblivious Transfer Schemes", Journal of Universal Computer Science, (2008) 14(3), pp.397-415.

[Chor et al.] Chor, B., Fiat, A., Naor, M.: "Tracing traitors"; Advances in cryptology-CRYPTO'94. Springer Berlin Heidelberg, 1994.

[Choy et al. 2005] Choy, K. L., Lee, W. B., Lau, H. C., Choy, L. C.: "A knowledge-based supplier intelligence retrieval system for outsource manufacturing"; Knowledge-based systems 18.1 (2005): 1-17.

[Cloud AEC 2011] Cloud A. E. C.: "Amazon web services"; Retrieved November 9 (2011): 2011.

[Fujisaki and Okamoto 1999] Fujisaki, E., Okamoto, T.: "Secure integration of asymmetric and symmetric encryption schemes"; Advances in Cryptology-CRYPTO'99. Springer Berlin Heidelberg, 1999.

[Goyal et al. 2006] Goyal, V., Pandey, O., Sahai, A., Waters, B.: "Attribute-based encryption for fine-grained access control of encrypted data"; Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006.

[Goyal et al. 2008] Goyal, V., Lu, S., Sahai, A., Waters, B.: "Black-box accountable authority identity-based encryption"; Proceedings of the 15th ACM conference on Computer and communications security. ACM, 2008.

[Green et al. 2011] Green, M., Hohenberger, S., Waters, B.: "Outsourcing the Decryption of ABE Ciphertexts"; USENIX Security Symposium. 2011.

[Hanaoka et al. 2005] Hanaoka, Y., Hanaoka, G., Shikata, J., Imai, H.: "Identity-based hierarchical strongly key-insulated encryption and its application"; Advances in Cryptology-ASIACRYPT 2005. Springer Berlin Heidelberg, 2005. 495-514.

[Kapadia et al. 2007] Kapadia, A., Tsang, P. P., Smith, S. W.: "Attribute-Based Publishing with Hidden Credentials and Hidden Policies"; NDSS. Vol. 7. 2007.

[Li et al. 2011] Li, J., Wang, Q., Wang, C., Ren, K.: "Enhancing attribute-based encryption with attribute hierarchy"; Mobile networks and applications 16.5 (2011): 553-561.

[Li et al. 2013] Li, J., Li, J., Chen, X., Jia, C., Lou, W.: "Identity-based Encryption with Outsourced Revocation in Cloud Computing". IEEE Transactions on Computers, 64(2): 425-437, 2015.

[Li et al. 2010] Li, J., Kim, K.: "Hidden attribute-based signatures without anonymity revocation". Information Sciences, 180(9): 1681-1689, Elsevier, 2010.

[Li et al. 2014] Li, J., Huang, X., Li, J., Chen, X., Xiang, Y.: "Securely Outsourcing Attribute-based Encryption with Checkability". IEEE Transactions on Parallel and

Distributed Systems, 25 (8): 2201-2210, 2014.

[Li et al. 2013] Li, J., Chen, X., Li, J., Jia, C., Ma, J., Lou, W.: "Fine-grained access control system based on outsourced attribute-based encryption"; Computer Security-ESORICS 2013. Springer Berlin Heidelberg, 592-609, 2013.

[Lin and Lo 2013] Lin, K. W., Lo, Y. C.: "Efficient algorithms for frequent pattern mining in many-task computing environments"; Knowledge-Based Systems 49 (2013): 10-21.

[Ma et al. 2013] Ma, X., Li, J., Zhang, F.: "Outsourcing computation of modular exponentiations in cloud computing". Cluster Computing 16(4): 787-796, 2013.

[Ostrovsky et al. 2007] Ostrovsky, R., Sahai, A., Waters, B.: "Attribute-based encryption with non-monotonic access structures"; Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007.

[Sahai 1999] Sahai, A.: "Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security"; Foundations of Computer Science, 1999. 40th Annual Symposium on. IEEE, 1999.

[Sahai and Waters 2005] Sahai, A., Waters, B.: "Fuzzy identity-based encryption"; Advances in CryptologyCEUROCRYPT 2005. Springer Berlin Heidelberg, 2005. 457-473.

[Shamir 1985] Shamir, A.: "Identity-based cryptosystems and signature schemes"; Advances in cryptology. Springer Berlin Heidelberg, 1985.

[Vecchiola et al. 2009] Vecchiola, C., Chu, X., Buyya, R.: "Aneka: a software platform for .NET-based cloud computing"; High Speed and Large Scale Scientific Computing (2009): 267-295.

[Wu et al. 2009] Wu, W., Mu, Y., Susilo, W., Huang X.: "Certificate-based Signatures Revisited". Journal of Universal Computer Science (2009) 15(8): 1684-1659.

[Wu et al. 2012] Wu, Z., Xu, G., Yu, Z., Yi, X., Chen, E., Zhang, Y.: "Executing SQL queries over encrypted character strings in the Database-As-Service model"; Knowledge-Based Systems 35 (2012): 332-348.

[Yakut and Polat 2012] Yakut, I., Polat, H.: "Estimating NBC-based recommendations on arbitrarily partitioned data with privacy"; Knowledge-Based Systems 36 (2012): 353-362.

[Yu et al. 2010] Yu, S., Wang, C., Ren, K., Lou, W.: "Achieving secure, scalable, and fine-grained data access control in cloud computing"; INFOCOM, 2010 Proceedings IEEE. Ieee, 2010.

[Zahariev, A. 2009] Zahariev, A.: "Google app engine"; Helsinki University of Technology (2009).