Flexible And Fine-Grained Access Control In Cloud Computing Using Hierarchical Based Encryption Scheme

[1] C. Muthu Pandian,
[1]Assistant Professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Tamilnadu, India

[2] K. Seenivasan
[2]Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Tamilnadu, India

Abstract

Cloud computing is one of the today's most exciting technologies due to its ability to reduce costs associated with computing while increasing flexibility and scalability for the computer process. While providing this convenience by this new technology, data owners worried about their outsourced sensitive data for sharing on the cloud servers. Still it causes many challenges for data security and access control. To keep the sensitive data secure against the distrusted servers, existing solutions follow attribute set based encryption scheme with hierarchy of users to achieve the flexibility, scalability and fine-grainedness in access control of data. However, in doing so, these solutions introduce the computation overhead on the key management. This paper addresses these challenges by combing the ABE with CP-ABE .We implement the key structure by reducing the redundant keys for the same set of users at the same level in dealing with the efficient and flexible access control for outsourced data in the cloud computing. This extensive experimental analysis shows that our proposed scheme is highly scalable, flexible and provable secure under existing security models.

Keywords *– Cloud Computing, Security and access control*

1. Introduction

Cloud computing is a new computing technology which can built on virtualization mechanism, distributed, parallel and utility computing along with service oriented architecture. For the past many years, cloud technology has been emerging as one of the most powerful paradigms in the IT industry, and has paying attention from both academia and industry. Cloud computing holds the assurance of providing computing as the additional utility after the other four utilities (water, gas, electricity, and telephone).

The benefits of cloud computing comprises of reduced costs and capital expenditures, increased functioning efficiencies, scalability, flexibility, urgent time to market, and so on. Different service-oriented cloud computing models have been proposed, including infrastructure (IAAS), platform (PAAS), and software (SAAS) as services. Plentiful commercial cloud computing systems have been built at different levels, e.g.,[3] Amazon's ec2, Amazon's s3, and IBM's[4]blue cloud are Iaas systems, while goggle app engine and yahoo pig are representative PAAS systems, and Google's apps[6] and [5] Salesforce's customer relation management (CRM) system belong to SAAS systems.

Along with these cloud computing systems, enterprise users no longer need to invest in hardware/software systems or hire it professionals to maintain these it systems, thus they save cost on it infrastructure and human resources; on the other hand, computing utilities provided by cloud computing are being offered at a relatively low price in a pay-as-you-use style.

1.1 Security Issues

Even though, [7] [8] [10] [13] the great profit brought by cloud computing technology are stimulating for IT industries, researchers in academics, and probable cloud users. The security problems in cloud computing become serious obstacles which, without being properly addressed, will prevent cloud computing broad applications and usage in the future. One of the important security concern is data security and privacy in cloud technology due to its Internet-based storage and management of data's. In cloud computing, users have to store their data to the cloud service provider for storage and business purposes, at the same time as the cloud service provider is usually a commercial activity which cannot be totally trusted.

Data represents an extremely important benefit for any business, and enterprise users will face serious consequences if its private data is disclosed to their

business competitors. Thus, cloud users at the top level need to ensure that their data are kept confidential to strangers, including the cloud provider and their potential competitors. This is the important data security requirement. Data secrecy is not the only protection requirement. [21] Flexible and fine-grained access control is also powerfully required in the service-oriented cloud computing model. A system of health-care information on a cloud is required to limit the access of protected medical records to eligible doctors and a customer relation management system running on a cloud may allow access of customer information to high-level executives of the company only. In these situations, access control of sensitive data is either required by legislation (e.g., HIPAA) or company regulations [19][17].

Access control is a classic security issue which dates back to the 1960s and various access control schemes have been proposed since then.[10] Among them, Bell-La Padula (BLP) [11] and Bi Baare two famous security models. To achieve flexible and well-grained access control, a number of mechanisms have been proposed more recently. Unfortunately, these mechanisms are only valid to systems in which data owners and the service providers are in the identical trusted domain. Because of the data owners and service providers are usually not in the same trusted domain in cloud computing, a new access control scheme employing attributed-based encryption [16] is proposed which adopts the so-called [17] key-policy attribute-based encryption (KP-ABE) to enforce well-grained access control. However, this mechanism has a demerit of short of flexibility in attribute management and lacks scalability in dealing with multiple-levels of attribute authorities. We note that in contrast to KP-ABE, cipher text-policy ABE (CP-ABE) turns out to be well suited for access control due to its expressiveness in describing access control policies.

2. Related Work

Computer systems traditionally have had closed, centrally managed security domains. Every entity that can take actions within such a system has one or more identities in that domain. The system grants or denies an entity's requests to access certain resources according to its access control policies and the authenticated identities of the requester.
The underlying assumption is that entities in the system already know each other. Therefore, trust can be easily established based on each other's identity. Further without obtaining a local identity, an entity will not be able to interact with the system and gain access to the system's resources.

There [14] is little chance for the clients to apply their own access control policies for their information, and decide accordingly whether the server is trustworthy enough so that sensitive information can be disclosed. The approach of automated trust negotiation differs from traditional identity-based access control systems mainly in the following aspects:

1. Trust between two strangers is established based on parties' properties, which are proven through disclosure of digital credentials. A digital credential is a verifiable, un-forgeable, digitally signed assertion by a credential issuer about the properties of the parties mentioned in the credential. A credential often contains a public key of one or more of the parties it mentions, so that those parties can prove that the credential de scribes them. Digital credentials can be implemented via X.509 certificates.

2. Every party can access control policies to control outsiders' access to their insightful resources. These resources can include benefits accessible from the Internet and role-based access control system, policy, credential and capability in capability based systems.

3. In the approaches to trust negotiation developed so far, two parties establish trust directly without involving trusted third parties, other than credential issuers. Since both parties have access control policies, trust negotiation can employ a peer-to-peer construction, in which a client and server are treated uniformly. Instead of a one-shot authorization and authentication, hope is established incrementally throughout a sequence of mutual credential disclosures. Less sensitive credentials are disclosed first. Later on, when a certain level of trust has been established, more sensitive credentials can be disclosed.

A number of cryptographic credential schemes and associated protocols have been developed to address these and other problems.[18] Oblivious signature based envelope, hidden credentials, and secret handshakes can be used to address the policy cycle problem. Oblivious Attribute Certificates (OACerts), private credentials, and anonymous credentials together with zero-knowledge proof protocols can be used to prove that an attribute satisfies a policy without disclosing any other information about the attribute. Certified input private policy evaluation (CIPPE) [20] enables A and B to determine whether A’s attribute values satisfy B’s policies without revealing additional information about A’s attributes or B’s policies.

CP-ABE is more intuitive as it is similar to traditional access control model where data is protected with access policies and users with credentials satisfying the policy are allowed access to it. The various CP-ABE schemes proposed the one proposed by Bethencourt et al is the most defending scheme. [2], we will refer this as a BSW,

is the most practical to date. It supports arbitrary strings as attributes, numerical attributes in keys and integer comparisons in policies and provides a means for periodic key refreshment. Furthermore, the authors have developed a software prototype with a friendly interface for combination in systems. However, BSW and other CP-ABE schemes are still far from being able to support the needs of modern enterprise environments, which require considerable flexibility in specifying policies and managing user attributes as well as better efficiency. Due to this, the fact that keys in current CP-ABE schemes can only support user attributes that are organized logically as a single set; i.e., users can use all possible combinations of attributes issued in their keys to satisfy policies.

Fuzzy-IBE gives rise to two interesting new applications. The first is an Identity-Based Encryption system that uses biometric identity. That can be viewed as a user's biometric, example for this is an iris scan, as that user's identity described by several attributes and then encrypt to the user using their biometric identity. Since biometric measurements are noisy, we cannot use existing IBE systems. However, the error-tolerance property of Fuzzy-IBE allows for a private key (derived from a measurement of a biometric) to decrypt a cipher text encrypted with a slightly different measurement of the same biometric.

Secondly, [15] Fuzzy IBE can be used for an application that we call this encryption as attribute based encryption. Through this application a party will wish to encrypt a document to all users that have a certain set of attributes. For example, in a computer science department, the chairperson might want to encrypt document to its entire systems faculty on a hiring committee. In this case it would encrypt to the identity {"hiring-committee","faculty","systems"}. Any user who has an identity that contains all of these attributes could decrypt the article. The advantage of using this Fuzzy IBE is that the document can be stored on an simple un trusted storage server instead of relying on trusted server to perform authentication checks before delivering a document.

3. Problem statement

In this section, we first present our HASBE scheme, which extends the ASBE algorithm with a hierarchical user structure. We then show how HASBE is applied for hierarchical user grant; data file creation, file access, user revocation, and file deletion.

Bilinear Maps: Let, G,G_1 be cyclic (multiplicative) groups of prime order. Let be a generator of. Then: $G\times G \rightarrow G_1$ is a bilinear map if it has the following properties:.

- Bilinearity: for all $u,v \in G$ and $a,b \in Zp$, $e(u^a,v^b) = e(u,v)^{ab}$
- Nondegeneracy: $e(g,g) \neq 1$.

G is called a bilinear group if the group operation and the bilinear map are both efficiently computable. In our HASBE scheme, a data encryptor specifies an access structure for a cipher text which is referred to as the cipher text policy. Only users with decryption keys whose associated attributes, specified in their key structures, satisfy the access structure can decrypt the cipher text.

Key Structure: We use a recursive set based key structures in where each element of the set is either a set or an element corresponding to an attribute. The depth of the key structure is the level [19] of recursions in the recursive set, similar to definition of depth for a tree. For a key structure with depth 2, members of the set at depth 1 can either be attribute elements or sets but members of a set at depth 2 may only be attribute elements. Consider the example shown in Figure. 1, where{Dept : DoD, Agency : DARPA,Position : Director, Level : 3}, {Position : Coordinator, Level : 6}} is a key structure of depth 2. It represents the attributes of a person who is both a director of level 3 for a unit and a coordinator of level 6 for another unit in the Defense Advanced Research Projects Agency (DARPA) of the Department of Defense (DoD). The key structure defines unique labels for sets in it. For key structures of depth 2, just an index of the sets at depth 2 is sufficient to uniquely identify the sets. Thus if there are m sets at depth 2 then a unique index where $1 \leq i \leq m$ is assigned to each set.

The set at depth 1 is referred to as set 0. Using this convention, a key structure of depth 2 can be represented as $A=\{A_0,A_1,\ldots..A_m\}$,where$A_0$is the set at depth 1 while A_i is the set at depth 2, for$1 \leq i \leq m$ In the key structure in Fig. 1.3,{Dept : DoD, Agency : DARPA}corresponds toA_0 , {Position : Director, Level : 3}and{Position : Coordinator, Level : 6}correspond to A_1 andA_2respectively

Individual attributes inherit the label of the set they are contained in and are uniquely defined by the combination of their name and their inherited label. For example, attribute Dept : DoD is defined as(0,Dept : DoD).When trying to satisfy a given policy, a user may only use attribute elements within a set, but may not combine attributes across the sets by default.

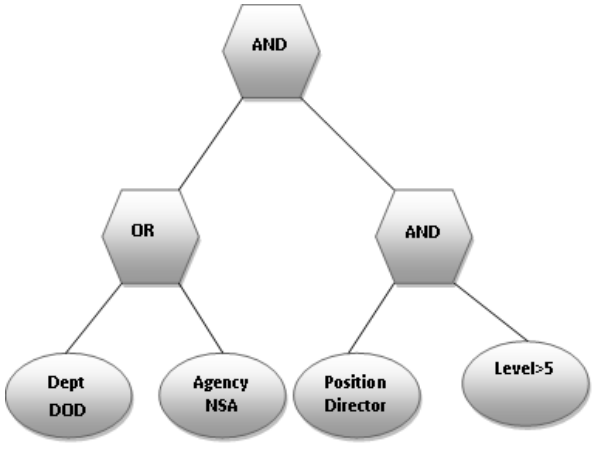

Figure 1.Example Access Structure

However, if the encryptor has designated translating node*s* in an access structure, users can combine attributes from multiple sets to satisfy the access structure, as will be explained later in the scheme construction as well as in.

Access Structure*:* In our scheme, we use the same tree access structure as in. In the tree access structure, leaf nodes are attributes and non leaf nodes are threshold gates. Each non leaf node is defined by its children and a threshold value. Let numx denote the number of children and k_x the threshold value of node. An example of the access tree structure is shown in Figure.1, where the threshold values for "AND" and "OR" are 2 and 1, respectively. The above access structure demands that only a director in DoD or NSA of level larger than 5 can access the data files protected by the access policy. In CP-ABE schemes, a person who has private keys corresponding to attributes on the key structure shown in Figure. 1 would be able to access the data files, which compromises the security of the access policy in Figure. 1. Such problems are effectively prevented using attribute-set-based encryption which forbids combining attributes across multiple sets [19].

Let T_x be the access structure rooted at node x and T be the access structure rooted at the root node. Without loss of generality, we consider key structure of depth 2,$A=\{A_0,A_1,\ldots..A_m\}$, where $A_i(0\leq i \leq m)$is the i^{th}attribute set and is the label. We say that satisfies T if and only if a function T(A) returns a nonempty set of labels. ThefunctionT (A) is computed recursively and will be introduced in the encryption algorithm later. A is said to satisfy T if it contains at least one set $A_i(0\leq i \leq m)$that has all the attributes needed to satisfy T and that the attributes belonging to multiple sets in A cannot be combined to satisfy T , except when there are designated translating nodes in T .[13] If node x is a translating node in T, then if the attribute elements used to satisfy the predicate represented by the sub tree rooted at x belong to a different set in A than those used to satisfy the predicates represented by the siblings of , the decrypting user is able to combine them to satisfy the predicate represented by the parent node.

Several functions are defined for the purpose of dealing with the access structure. We define parent(x) as the parent node of xand index(x) as the index number of node x. The function att(x) is defined only if xis a leaf node and denotes the attribute associated with the leaf node x in the tree.

4. System Model

As depicted in Figure.2, the cloud computing system under consideration consists of five types of parties: a cloud service provider, data owners, data consumers, a number of domain authorities, and a trusted authority. The cloud service provider manages a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them. Each data owner/consumer is administrated by a domain authority.

A domain authority is managed by its parent domain authority or the trusted authority. Data owners, data consumers, domain authorities, and the trusted authority are organized in a hierarchical manner as shown in Figure. 2. The trusted authority is the root authority and responsible for managing top-level domain authorities.

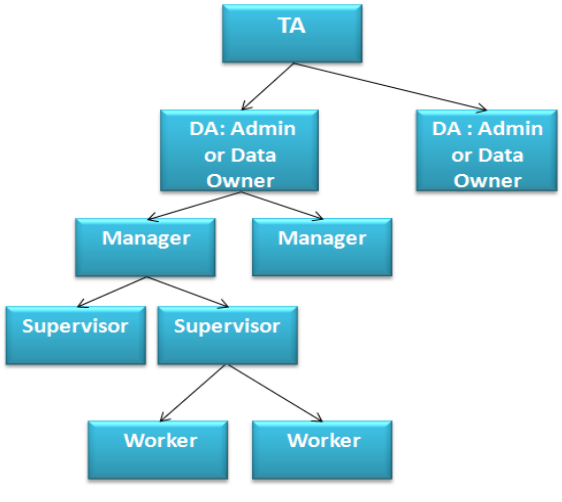

Figure 2. System Model

Data owners/consumers may correspond to employees in an organization. Each domain authority is responsible for managing the domain authorities at the next level or the data owners/consumers in its domain. In our system,

neither data owners nor data consumers will be always online. They come online only when necessary, while the cloud service provider, the trusted authority, and domain authorities are always online. The cloud is assumed to have abundant storage capacity and computation power. In addition, we assume that data consumers can access data files for reading only. Each top-level domain authority corresponds to a top-level organization, such as a federated enterprise, while each lower-level domain authority corresponds to a lower-level organization, such as an affiliated company in a federated enterprise. The access permissions of each user is set by the data owner as shown in the Figure 3.

Number of Levels In the Organization	Attributes / Users	Jewel Name	Dealer Name	Buying Date	Total Grams	Sales Rate	Carat Type	Buying Date	Original Data File: Jewel Image And Description	
Level 1	Administrator or Data Owner	YES	YES	YES	YES	YES	YES	YES	YES	YES
Level 2	Manager	YES	YES	NO	YES	YES	YES	YES	YES	YES
Level 3	Supervisor	YES	NO	NO	YES	YES	YES	YES	YES	YES
Level 4	Worker	YES	NO	NO	YES	YES	YES	NO	YES	YES

YES - Permission granted for access the attributes NO- Permission not granted for access the attributes

Figure 3. Access control level in an organization

5. Result Analysis

The following graphs are showing the result analysis of Hierarchical attribute based encryption scheme using various parameters. These graphs are showing using the following metrics such as number of key generations, enabling of attributes and key generation time.

5.1 Number of Key Generation

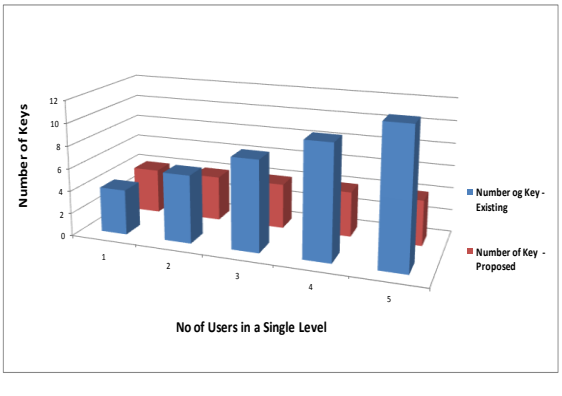

Figure 4. Number of key Generations – Graph.

In the above graph, the number of keys is taken as a metric. The x-axis shows Number of users in a single level of organization and y-axis shows the number of keys. In the existing system, the number of keys generated at a single level is more. If the attributes of a single user is increased, then the number of keys at a single level for a single user automatically increased. In the existing system, the duplicate keys i.e the different keys are generated for the same attribute is dropped. So the keys at a single level will be greatly reduced.

5.2 Enabling of attributes

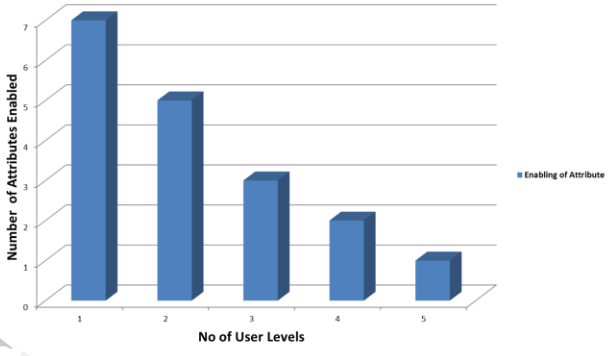

Figure 5. Enabling of attributes – Graph.

In the Figure 5, the number of attributes enabled at each level is taken as a metric. The x-axis shows the number of user levels and y-axis shows the number of attributes enabled at a each level. In the top level (root node) or administrator of the organization has full control of all the attributes since all the attributes are enabled for that user.

In the descendent levels, the attributes are disabled according to the access policies defined by the data owner. So the number of attributed enabled is linearly increased according to the number of levels in an organization.

5.3 Key Generation Time

In the Figure 6, the key generation time is taken as a metric. The x-axis shows the number of attributes and y-axis shows the key generation time of each user at different levels in the organization. In the existing system, the total time to generate the keys are increased because of all same attributes has different keys. The existing systems are showed in the blue bars in the figure.

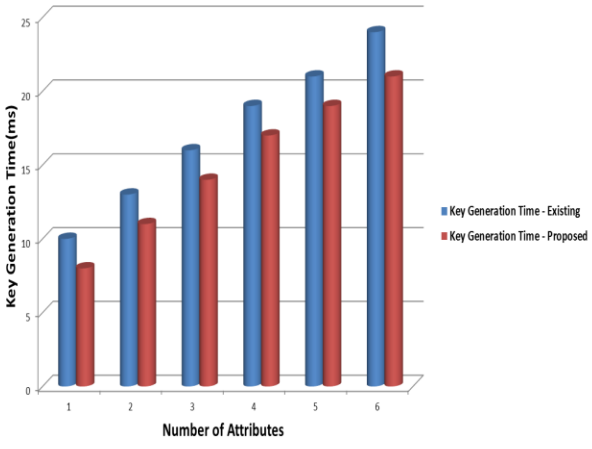

Figure 6. Key Generation Time – Graph.

In the proposed system, the duplicate keys are reduced greatly by having a common key for same attributes used for various users in the organization. This is showed in the red bars in the figure.

6. Conclusion

The scheme introduced for realizing scalable, flexible, and fine-grained access control in cloud computing. The scheme seamlessly incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. This scheme is not only supports compound attributes due to flexible attribute set combinations, but also achieves efficient user revocation because of multiple value assignments of attributes. We formally proved the security of scheme based on the security of CP-ABE. Finally, we improve the flexibility, scalability and fine grained access control of this system by reducing the number of keys generated for same attributes in the same level of organization. We implemented the proposed scheme, and conducted comprehensive performance analysis and evaluation, which showed its efficiency and advantages over existing schemes.

REFERENCES

[1] R. Buyya, C. ShinYeo, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Comput. Syst.*, vol. 25, pp. 599–616, 2009.
[2] Amazon Elastic Compute Cloud (Amazon EC2) [Online]. Available: http://aws.amazon.com/ec2/
[3] Amazon Web Services (AWS) [Online]. Available: https://s3.amazonaws. Com/
[4] R. Martin, "IBM brings cloud computing to earth with massive new data centers," *InformationWeek* Aug. 2008 [Online]. Available: http:// www.informationweek.com/news/hardware/data_centers/209901523
[5] Google App Engine [Online]. Available: http://code.google.com/appengine/
[6] K. Barlow and J. Lane, "Like technology from an advanced alien culture: Google apps for education at ASU," in *Proc. ACM SIGUCCS User Services Conf.*, Orlando, FL, 2007.
[7] B. Barbara, "Salesforce.com: Raising the level of networking," *Inf. Today*, vol. 27, pp. 45–45, 2010.
[8] J. Bell, Hosting Enterprise Data in the Cloud Part 9: InvestmentValueZetta, Tech. Rep., 2010.
[9] A. Ross, "Technical perspective: A chilly sense of security," *Commun. ACM*, vol. 52, pp. 90–90, 2009.
[10] D. E. Bell and L. J. LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation The MITRE Corporation, Tech. Rep., 1976.
[11] K. J. Biba, Integrity Considerations for Secure Computer SytemstheMITRE Corporation, Tech. Rep., 1977.
[12] H. Harney, A. Colgrove, and P. D. McDaniel, "Principles of policy in secure groups," in *Proc. NDSS*, San Diego, CA, 2001.
[13] P. D. McDaniel and A. Prakash, "Methods and limitations of security policy reconciliation," in *Proc. IEEE Symp. Security and Privacy*, Berkeley, CA, 2002.
[14] T. Yu and M. Winslett, "A unified scheme for resource protection in automated trust negotiation," in *Proc. IEEE Symp.Security and Privacy*, Berkeley, CA, 2003.
[15] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in *Proc. ACM Conf. Computer andCommunications Security (CCS)*, Alexandria, VA, 2005.
[16] V. Goyal, O. Pandey, A. Sahai, and B.Waters, "Attibute-based encryption for fine-grained access control of encrypted data," in *Proc. ACMConf. Computer and Communications Security (ACM CCS)*, Alexandria, VA, 2006.
[17] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM 2010*, 2010, pp. 534–542.
[18] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebasedencryption," in *Proc. IEEE Symp. Security and Privacy*, Oakland,CA, 2007.
[19] R. Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practicallymotivated enhancement to attribute-based encryption," in *Proc.ESORICS*, Saint Malo, France, 2009.
[20] A. Sahai and B. Waters, "Fuzzy identity based encryption," in *Proc. Acvances in Cryptology—Eurocrypt*, 2005, vol. 3494, LNCS, pp.457–473.
[21] G.Wang, Q. Liu, and J.Wu, "Hierachicalattibute-based encryption for fine-grained access control in cloud storage services," in *Proc. ACMConf. Computer and Communications Security (ACM CCS)*, Chicago, IL, 2010.