American Journal of Software Engineering and Applications
2018; 7(1): 1-14
http://www.sciencepublishinggroup.com/j/ajsea
doi: 10.11648/j.ajsea.20180701.11
ISSN: 2327-2473 (Print); ISSN: 2327-249X (Online)

Review Article

Research on Access Control in Cloud Storage System: From Single to Multi-Clouds

Zhijie Fan[1, *], Ya Xiao[1, *], Chunmei Wang[2, *], Bing Liu[1, *]

[1]Electronics and Information Engineering School, Tongji University, Shanghai, China

[2]Ping An Insurance (Group) Company of China, Shanghai, China

Email address:
aaronzfan@126.com (Zhijie Fan), 1710053@tongji.edu.cn (Ya Xiao), goldey@126.com (Chunmei Wang), liu_bing_2010@sina.com (Bing Liu)

[*]Corresponding author

To cite this article:
Zhijie Fan, Ya Xiao, Chunmei Wang, Bing Liu. Research on Access Control in Cloud Storage System: From Single to Multi-Clouds. *American Journal of Software Engineering and Applications*. Vol. 7, No. 1, 2018, pp. 1-14. doi: 10.11648/j.ajsea.20180701.11

Received: February 24, 2018; **Accepted**: March 21, 2018; **Published**: April 13, 2018

Abstract: Implementation of access control in cloud storage system is the essential method to protect users' data from revealing sensitive information. The paper mainly investigates key technologies of access control in cloud storage system, including intra cloud and among multi-clouds. Firstly, we discuss about the focuses in recent researches and challenges of access control in cloud storage system. The access control researches here refer to cipher-text and cross-domain access control in cloud storage system. The key technologies introduce Ciphertext-Policy Attribute-Based Encryption algorithm(CP-ABE), ontology based attributes mapping, algebra based policies integration, solutions for identification, access authorization and identity federation. And the status of these fields is described next. At last, we concluded this paper and proposed some directions in the future work of access control research in cloud storage system. This paper can help to understand the key technologies of access control in cloud storage and helpful in the future researches.

Keywords: Cloud Storage, Access Control, CP-ABE, Ontology, Identification

1. Introduction

With the rapid development of cloud computing, lots of applications on cloud storage are applied in industries. Cloud storage [1] is the realization of virtualized storage in demand. It extends and develops the concept of cloud computing, named data storage as a service (DaaS). At present, applications of cloud storage consist of storage back-up, data archiving, application data storage and others. Cloud computing can offer powerful compute and storage capacity to users, easy to access and low-cost. Its unique framework also brings some security and privacy concerns for users' data is no longer stored in local but in cloud. So as to protect the security of data, cloud services provider (CSP) must make sure only authorized entities can access the data. The cloud environment is un-trusted, even the CSPs, so it is also necessary to avoid leaking sensitive information to CSPs. To keep the unauthorized entities from accessing the data, it's important to implement access control in cloud storage system, and the data should be encrypt at the same time to avoid the access by CSPs. Hence one of the researches in cloud storage access control is how to implement access control on large cipher-text in cloud.

As the extension of cloud storage services, signal CSP storage system cannot meet the needs of users, so they store different resources in different clouds, there comes out the researches on managing multiple CSPs with one signal platform or interface [2, 3]. In multiple CSPs, resources are stored in different security domains (in this paper, we assume a cloud is a security domain, so different security domains means multiple clouds), certainly there are different access control policies among these security domains. Under this circumstance, we must unify and coordinate each security domain and build an accordance access control policy before accessing resources cross domains. Certainly the methods used in cross-domain access control must fit for cloud environment. With regards to the storage environment of

cooperating among multi-clouds, how to realize access control policy among different security domains has become another challenge in access control researches of cloud storage system.

The paper mainly research on some related issues and key technologies in the two aspects we proposed above on access control in cloud storage system, including the progress of researches in this field. The rests of the paper are constructed as follows: In section 2 and 3, we describe some related key technologies in the two aspects; the progress of researches in this filed will be depicted in section 4; in section 5, we conclude the whole paper and propose some advice in the future work of access control research in cloud storage system.

2. Key Technologies in Cipher-Text Access Control in Cloud Storage

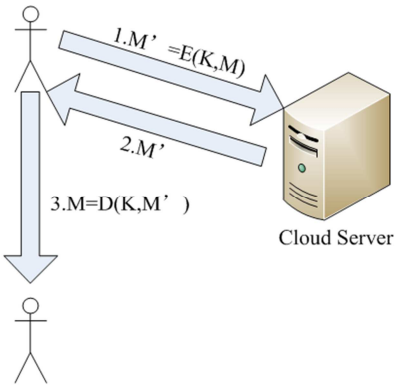

Figure 1. *A simple resolution for cipher-text access control in cloud storage system.*

The simplest way to resolve large cipher-text access control problem in cloud storage system is the user uses his own key to encrypt data and upload to the cloud servers, in this way he needs to retrieve and decrypt the data on his own and finally transmit the data to the shared users, the process is shown in Figure 1. But there are three serious problems in this scheme: First, users must do most of the computation and communication work by themselves; Second, users should manage their own keys, once lost the whole security would be destroyed; third, the CSPs can't forward the data directly, which is a great weakness.

Obliviously, this method is not fit for cloud storage environment, and most researchers are using cryptology based methods to implement cipher-text access control in cloud storage system recently.

Cipher mechanism [4] is to encrypt data and only the authorized users can decrypt it. The resource owner encrypts the resources before they are stored, and realize access control by controlling the user's access of keys. With this access control method, the confidentiality of data can be protected in un-trusted environment. This method is mainly used in sensitive data and data with many interrelationships of subjects and objects. There are several key technologies: hierarchical key generate and distribute policy enforcement based access control policy [5], attribute-based encryption algorithm (including key-policy attribute-based encryption (KP-ABE) [6] and ciphertext-policy attribute-based encryption (CP-ABE) [7] [8]), Proxy re-encryption [9]. CP-ABE algorithm supports users to set access rules and realize fine-grained access control where the un-trusted third party will not be introduced, therefore CP-ABE based schemes are researched and applied in cloud storage system most popularly, and we will emphatically depict the CP-ABE algorithm in the following part.

2.1. CP-ABE

CP-ABE was firstly proposed based on attribute-based encryption (ABE) [10] to solve the problem that ABE can't support flexible access control policy. ABE uses access structure to express access policy with bilinear pairing, to obtain security by using some math challenges and hypothesizes. The CP-ABE algorithm adopts general group model in complexity hypothesis, and supports complicated policy, in addition the access policy can be constructed by sender. Therefore this algorithm is more suitable to be applied in applications who need fine-grained access control, such as cloud storage. The mechanism of CP-ABE is showed in Figure 2.

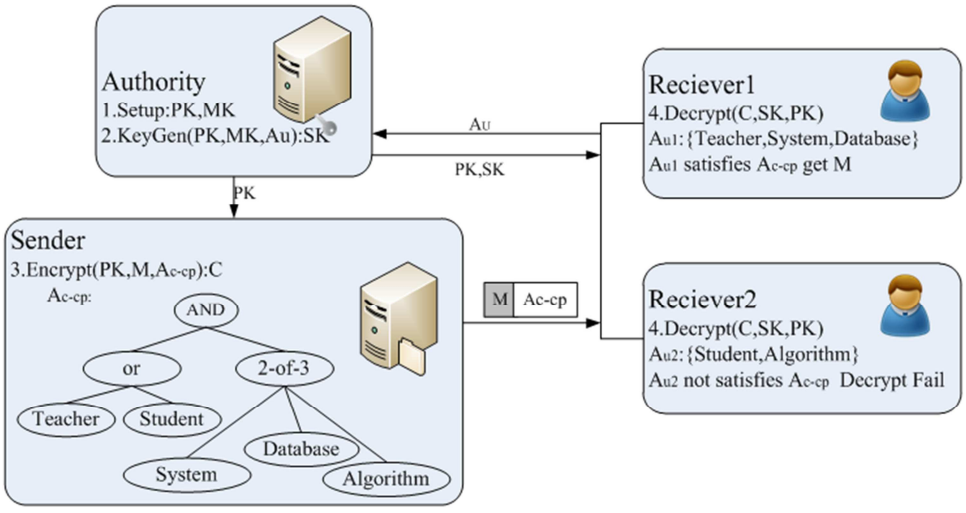

Figure 2. *The mechanism of CP-ABE.*

Three parties are participated in CP-ABE algorithm, including authority, sender and receiver. The four main steps are as follow:

a) Authority runs Setup algorithm, and generate a public key PK and a main key MK.
b) Authority runs Keygen algorithm, and generate a secret key SK. This operation will involve receivers' attributes set AU into SK, then distribute the SK to receivers in a secure channel.
c) Sender runs Encrypt algorithm, and encrypt the message M into cipher text C. The cipher text C includes access policy Ac-cp which is described in tree structure.
d) Receiver can run Decrypt algorithm to decrypt cipher text C and get message M only when his attributes set matches the access policy Ac-cp.

2.2. CP-ABE Based Cloud Storage Access Control Framework

We have discussed the main steps in CP-ABE algorithm above, including system setup, data publication and retrieve. At the moment several cloud access control models based on CP-ABE have been proposed, and the general framework is shown in Figure 3.

The framework in Figure 3 shows that three parties are involved: sender, receiver and CSP, the access policy T is described as a tree structure. The progresses of system setup, data publication and retrieve are similar as the mechanisms of CP-ABE discussed above. But there are several differences in this framework. The sender needs to generate and distribute keys, then sends the cipher text C to the cloud servers for storing, and the receiver can retrieve and decrypt the message in cloud servers.

The fifth step named user revocation completed by sender in Figure 3 is one of the most important processes in CP-ABE algorithm.

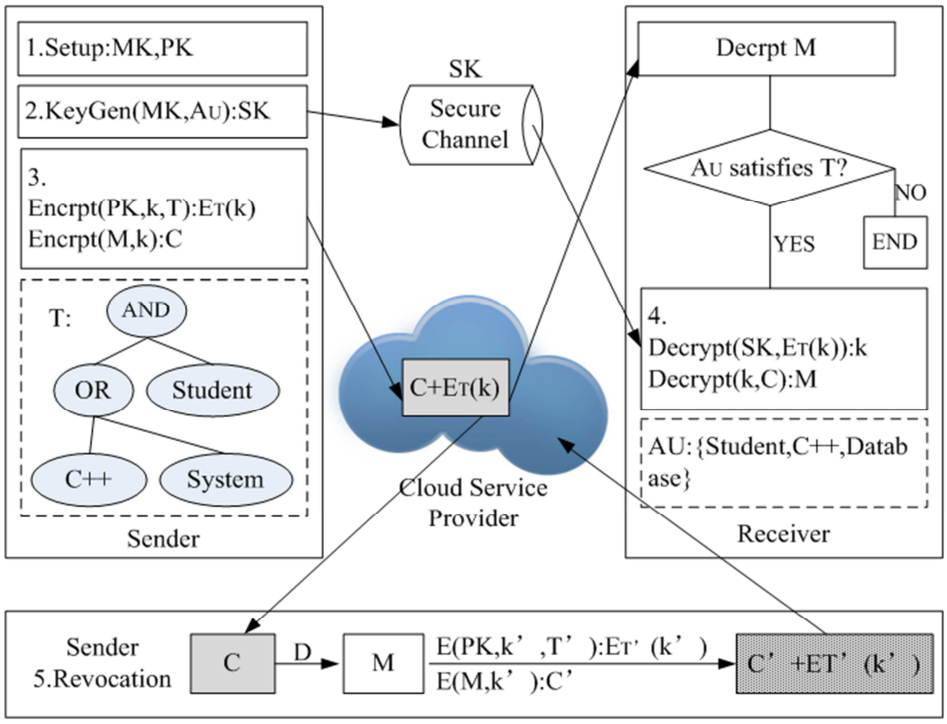

Figure 3. *CP-ABE based cloud storage access control framework.*

CP-ABE algorithm is considered as the most suitable technology which can be applied in cipher text access control in cloud storage system, for the reason that the data owner can construct access policy to realize fine-grained access control in the open network even policy evaluation can be done in cipher text. But user revocation requires user to retrieve, re-encryption and re-publication a large amount of data as we described above, which will consume a lot of computations and bandwidths in cloud computing environment. So CP-ABE algorithm should be optimized and improved so that it can be more suitable when applied in access control in cloud storage system.

3. Key Technologies in Cross-Domain Access Control in Cloud Storage

The existing technologies and solutions on cross-domain cloud storage access control mainly cover cloud identification, cloud access authorization, cloud identity federation et al. The access control model in cross-domain cloud storage system consists of two main schemes: Role-based access control (RBAC) [11, 12] and Attribute-based access control (ABAC) [13, 14]. In RBAC based scheme, the authorization mechanism is static, and also low support for fine-grained access control, so it's not a good choice to be applied into open

network like cloud storage system. While ABAC model was proposed to solve access control issues in distribution network, and fine-grained access control is well supported in this model. Therefore ABAC model is more suitable to be applied in cloud storage system. We will discuss some issues and solutions of access control based on ABAC model and the main technologies in cross-domain cloud storage.

3.1. ABAC Model

ABAC model uses attributes to define privileges, and three kinds of attributes are related to access control: subject, resource and environment attributes. Simple descriptions of these three attributes are as follows, and Figure 4 is an ABAC access control model.

a) Subject attributes: A subject is an entity that can operate resource, and it can be a user, an application or a process et al. Each subject has many attributes to identify and describe a subject, and these attributes can be ID, name, address and title et al.
b) Resource attributes: A resource is an entity that operated by subjects, and it can be a web server or document et al. Each resource has many attributes to identify and describe a resource, which can be used as access control policy evaluation.
c) Environment attributes: To describe various environments when a subject is accessing a resource, such as technical, operational, situational and context environment.

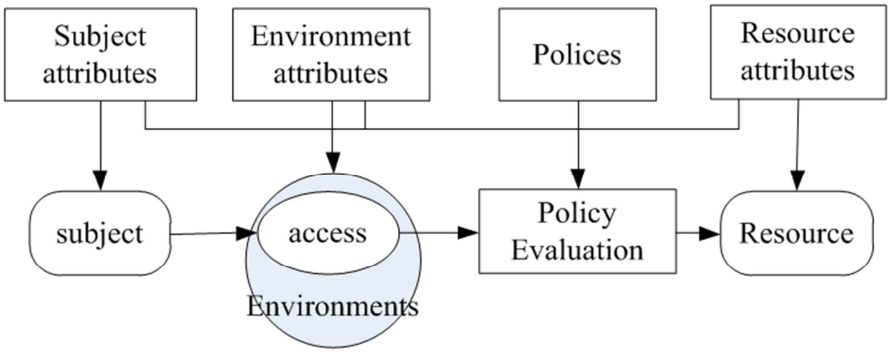

Figure 4. *ABAC access control model.*

ABAC model completes access control by formulating and evaluating policies as shown in Figure 4. These policies can be any association of attributes, so that ABAC model can reach more flexible and fine-grained access control. In order to have a further discussion, we will give out a simple formalization expression of ABAC model.

a) Let *S*, *R*, *E* express subjects set, resources set and environments set respectively, and *ATT_S, ATT_R, ATT_E* are subject attributes set, resource attributes set and environment attributes set, and SA_l, RA_m, EA_n are defined attributes of subject, resource and environment. The detail expression as denoted in (1).

$$\begin{aligned} ATT\_S &= \{SA_1, SA_2, \cdots, SA_l\} \quad 1 \le l \le L \\ ATT\_R &= \{RA_1, RA_2, \cdots, RA_m\} \quad 1 \le m \le L \\ ATT\_E &= \{EA_1, EA_2, \cdots, EA_n\} \quad 1 \le n \le L \end{aligned} \tag{1}$$

b) The three attribute sets' value domains are indicated as *D(SA), D(RA), D(EA)*; and use a triple *(s, r, e)* to describe one access, including subject s, resource r and environment e. They are described in (2) respectively:

$$\begin{aligned} s &= \{sa_1, \cdots, sa_l\} \in D(SA_1) \times \cdots \times D(SA_l), s \in S \\ r &= \{ra_1, \cdots, ra_m\} \in D(RA_1) \times \cdots \times D(RA_m), r \in R \\ e &= \{ea_1, \cdots, ea_n\} \in D(EA_1) \times \cdots \times D(EA_n), e \in E \end{aligned} \tag{2}$$

c) Access policy is defined as (3):

$$Policy = \{(s_1, r_1, e_1), (s_2, r_2, e_2), \cdots, (s_l, r_m, e_n)\} \tag{3}.$$

According to the definitions above, if *(s, r, e)* $\in$*Policy*, it means that subject s is allowed to access resource r under the environment e, or it indicates that require is denied.

Due to the semantic of attributes and policies may be heterogeneous among each domain under environments of multiple security domains, it's necessary to implement attributes mapping and policies integration in cross-domain access control when applying ABAC model in multi-clouds storage. So far, many researchers [15-17] have proposed ontology based method to realize attributes concepts mapping between different domains, and other researchers [18-20] use algebra to integrate policies. The two key technologies are introduced as follows.

3.2. Ontology Based Attributes Concepts Mapping

Ontology [21] is a formalized standard description of a shared conceptual model. The main differences between description in ontology and traditional knowledge are that the contents, like concepts, attributes, constraint conditions, interrelationships et al, described in ontology can be understood by computer and the implicit information can be expressed by ontology. In this way, a common understanding can be established between man and computer, and the descriptions can be reuse directly. Ontology Language is a formal language to structure ontology, this language can encode knowledge in particular field, including support inference rules to process the knowledge. The Web Ontology Language (OWL) [22, 23] can be used to simulate the

elements like users, resources, strategies, attributes, constraint conditions, et al in cloud environment, and to map the operations on access control into operations on ontology.

The formalization of concepts and relationships between concepts can be expressed in ontology, so it can be used to resolve the issue of attributes semantic matching in ABAC model with multi-domains. The attributes matching of ontology concepts in different domains can use similar calculating method. In this method, the similarity between two concepts is computed first, and then it is compared with the pre-defined threshold to determine the relationship between two concepts. The similarity calculating includes computing the similarity of name, instance, attribute and integration.

a) Name similarity calculating: before computing, the acronyms in names need to be converted into original words according the domain vocabulary, then use Levenshtein edit distance [24] to compute concepts' name similarity. The name similarity of concept C_1 and C_2, and $Sim_{name}=(C_1, C_2)$ is defined as (4)

$$Sim_{name}(C_1, C_2) = \max(0, \frac{\min(|C_1|, |C_2|) - edit(C_1, C_2)}{\min(|C_1|, |C_2|)}) \tag{4}$$

Where $edit\ (C_1, C_2)$ is the equation of Levenshtein edit distance, and $min(|C_1|, |C_2|)$ is the minimum character number contained in two strings.

b) Instance similarity calculating: to compute concepts similarity within concepts' instances. It's based on the theory: if two concepts' instances are all the same, then the two concepts are equal. I_1 and I_2 are instances sets of concepts C_1 and C_2 respectively, and instance similarity is defined as (5), where $P(C_1 \cap C_2)$ is the probability of common instances appear in both C_1 and C_2, and $P(C_1 \cup C_2)$ is the probability of all the instances appear in C_1 and C_2.

$$Sim_{instance}(C_1, C_2) = \frac{P(C_1 \cap C_2)}{P(C_1 \cup C_2)} \tag{5}$$

The similarity value is between 0 and 1, $Sim_{instance}(C_1, C_2) \in [0, 1]$, minimum value 0 represents two concepts are complete uncorrelated, while maximum value 1 means two concepts are totally the same.

c) Attribute similarity calculating: concept's attribute has factors like name and data type. Here we use these two factors to compute attribute similarity. a_i is each attribute of concept C_1, and b_j is each attribute of concept C_2, for every factor in attribute, the similarity is calculated as (6), where a is adjust divisor.

$$Sim_{attname/type}(a_i, b_j) = \frac{P(a_i \cap b_j)}{P(a_i \cap b_j) + aP(a_i - b_j) + (1-a)P(b_j - a_i)} \tag{6}$$

The attribute similarity is defined as (7), where $w_{attname}$ and w_{type} are the weights of different factors in attribute.

$$Sim_{att}(a_i, b_j) = w_{attname} Sim_{attname}(a_i, b_j) + w_{type} Sim_{type}(a_i, b_j) \tag{7}$$

Assume that there are n attributes between concepts C_1 and C_2, the attribute similarity of C_1 and C_2 is defined as (8).

$$Sim_{att}(C_1, C_2) = \frac{\sum_{k=1}^{n} w_k Sim_{att}(a_i, b_j)}{\sum_{k=1}^{n} w_k} \tag{8}$$

d) Integration similarity calculating: sum up the name similarity, instance similarity and attribute similarity of concepts C_1 and C_2, and get integration similarity, as in (9).

$$Sim(C_1, C_2) = w_{name} Sim_{name}(C_1, C_2) + w_{instance} Sim_{instance}(C_1, C_2) + w_{att} Sim_{att}(C_1, C_2) \tag{9}$$

e) Finally, make a compare between integration similarity and pre-defined threshold, and determine the relationship of two attribute concepts.

The XACML based access control system introduced in chapter 3.5 will use this attributes concepts mapping technology.

3.3. Algebra Based Policies Integration

Using algebra system to describe, infer and calculate the attribute based access control policy integration in cross domain environment is an effective way to solve policy conflict and integration. This section will describe a model based on a classical access control policies integration algebra model proposed by Bonatti et al. [25]. The main idea is to consist of object, subject and action into a triple authorization item, and use operators like intersection, union and difference to describe different access control policy integration method. The formation expression of policy is described in section 3.1. Policy A and policy B are access policies of domain A and domain B, and the operators of intersection, union and difference are respectively represented as $\otimes$, $\oplus$ and $\perp$. The formulation expressions of policy integration are shown in (10), (11) and (12)

$$Policy1 \otimes Policy2 = \{(s, r, e) \mid (s, r, e) \in Policy1 \wedge (s, r, e) \in Policy2\} \tag{10}$$

Equation (10) means that if (s, r, e) satisfies $Policy1$ and $Policy2$, then the access requirement is allowed.

$$Policy1 \oplus Policy2 = \{(s, r, e) \mid (s, r, e) \in Policy1 \vee (s, r, e) \in Policy2\} \tag{11}$$

Equation (11) means that if *(s, r, e)* satisfies *Policy1* or *Policy2*, then the access requirement is allowed.

$$Policy1 \perp Policy2 = \{(s,r,e) \mid (s,r,e) \in Policy1 \wedge (s,r,e) \notin Policy2\} \tag{12}$$

Equation (12) means that if *(s, r, e)* satisfies *Policy1* but not *Policy2*, then the access requirement is allowed.

Here we use a simple example to illustrate this access control policy integration. Assume that there are two cloud applications *A* and *B*, and attributes' semantic has coordinated in two clouds. The access policies in two domains are: *PolicyA*, users whose credit is greater than 0.7 and identity as members, can read the files which security level is not higher than 2; *PolicyB*, users whose credit is greater than 0.8 and identity as members, can read the files which security level is not higher than 3. So *PolicyA* and *PolicyB* can be expressed as (13) and (14).

$$PolicyA = \{[< s_1, s_2 >, r, e] \mid s_1 > 0.7, \; s_2 = member, r \le 2, e = read\} \tag{13}$$

$$PolicyB = \{[< s_1, s_2 >, r, e] \mid s_1 > 0.8, \; s_2 = member, r \le 3, e = read\} \tag{14}$$

The result of two policies integration is (15).

$$PolicyA \otimes PolicyB = \{[< s_1, s_2 >, r, e] \mid s_1 > 0.8, \; s_2 = member, r \le 2, e = read\} \tag{15}$$

3.4. OAuth Cross-Domain Identification

The OAuth based cross-domain identification [26][27] is one of the most widely used cloud identification methods. OAuth is an open standard supports cross domain access, which means an application in one domain can access that in other domain. The enterprise can share private resources to users in other cloud without exposing users' authentication information. Users can use a third party application to access resources in one website with no need to provide names and passwords to the application. The implement process is show in Figure 5.

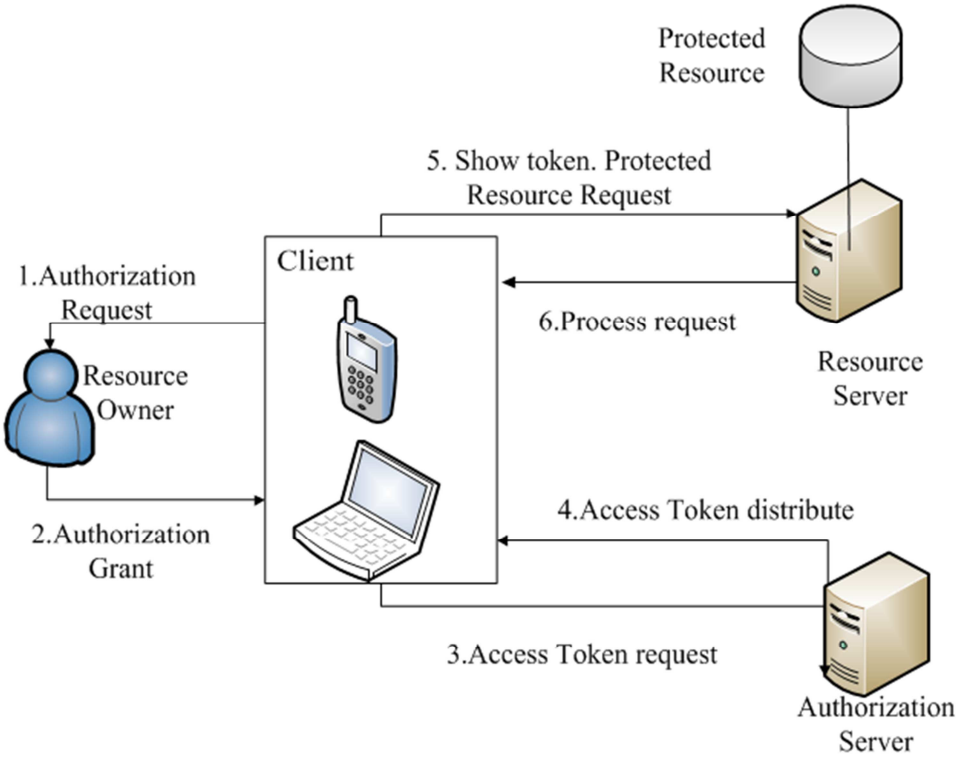

Figure 5. *The implement process of OAuth.*

The roles are defined as follows.

a) Resource Owner: users who can authorize other applications to store or access protected resources.
b) Resource Server: the server who stores the protected resources, to accept or deny the access request from application by analyzing the access token.
c) Client: the applications who access or store the protected resources on behalf of the resource owners.
d) Authorization Server: issue the access token to the client server after authorizing the resource owner and verifying the permission.

OAuth use an authorization layer to part client from resource owner. After the client getting the user's authorization, it can obtain an access token instead of username and password from authorization server. The client can use access token to store or access the protected resources, which assign the information like range and time.

3.5. XACML Based Access Authorization

The existing authorization methods are based on particular application authorization models, but for multiple application access authorizations, it is hard for these models to descript. A normalized language, access authorization method and execute policy should be proposed for different applications to establish a general authorization standard. The authorization standard is based on policies and rules, which decided by user roles and duties. The eXtensible Access Control Markup Language (XACML) [27][28] is a suitable standard in this circumstance. XACML, approved by OASIS, is a general access control language for policy management and access decision and supports general policy languages like XML, mainly used to realize access control for resources. As an access control standard, XACML not only have a policy language model, but also a policy management and access model, which is suitable for the environment with multiple domains and applications like cloud. XACML has a transplantable and standard method to describe access control entities and attributes, and provides a more fine-granted access control than simply refuse or authorize. The frame and processes of XACML access control are described in Figure 6.

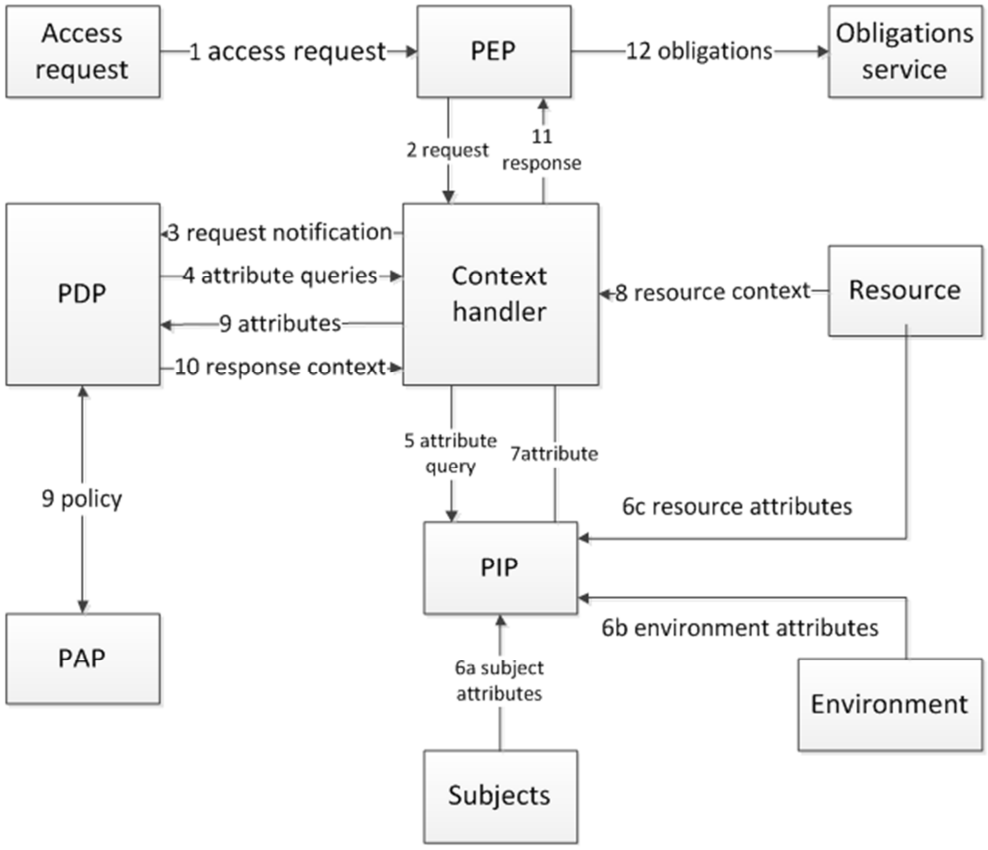

Figure 6. *Frame and processes of XACML access control.*

The usages of components are as follows.

PEP: Policy enforcement point, the system entity that performs access control, by making decision requests and enforcing authorization decisions.

PIP: Policy information point, the system entity that acts as a source of attribute values.

PDP: Policy decision point, the system entity that evaluates applicable policy and renders an authorization decision.

PAP: Policy administration point, the system entity that creates a policy or policy set.

Context handler: The system entity that converts decision requests in the native request format to the XACML canonical form, coordinates with PIP to add attribute values to the request context, and converts authorization decisions in the XACML canonical form to the native response format.

The processes of handling requests in XACML can be divided into 6 steps.

a) Access request sends to PEP. PEP gets the attributes of object, subject, environment and operant behavior from request and sends them to context handler.
b) Context handler converts the request into XACML canonical form and sends to PDP. PDP sends attributes query request to handler.
c) Handler delivers the request from PDP to PIP. PIP will query the attributes' information and return it to handler.
d) Handler sends the attributes' information to PDP after receiving it. PDP execute the policy which provided by PAP.
e) PDP sends the decision result of authorization to context handler.
f) Handler converts the result into native response format which can be recognized by PEP, and PEP handle the access request, like allow or deny.

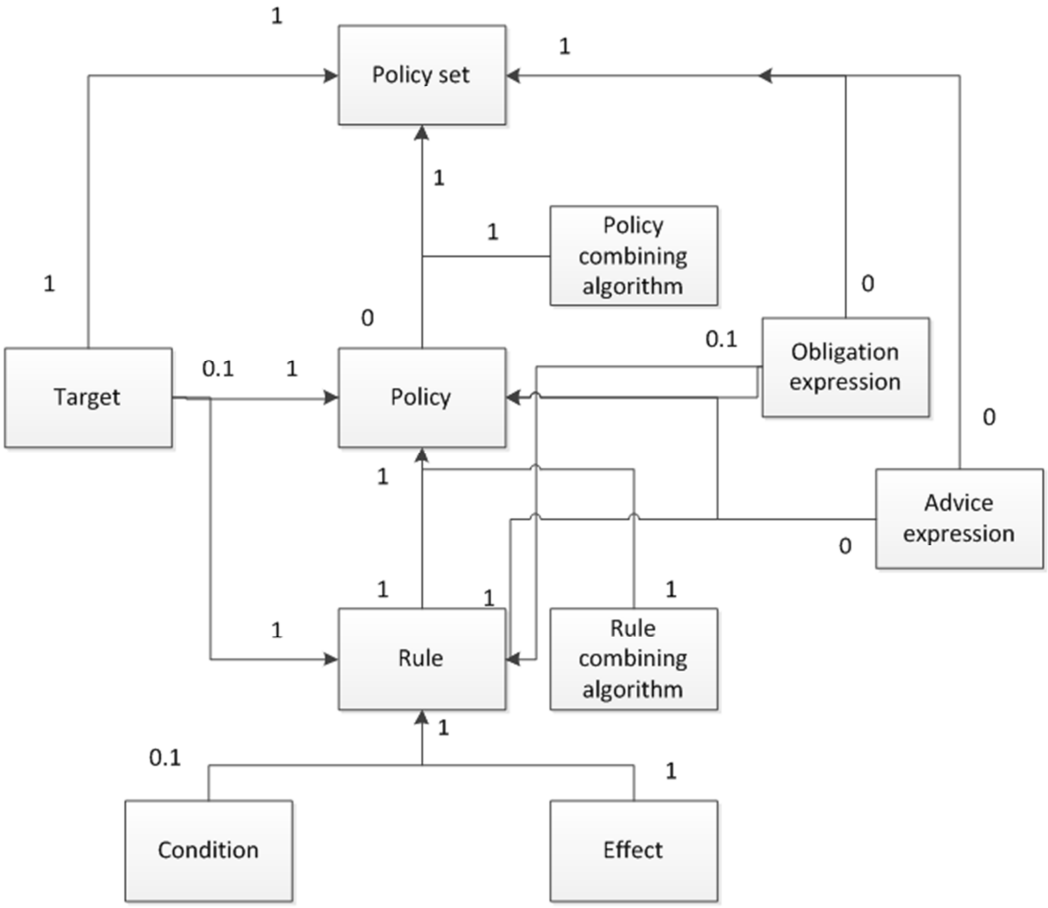

Figure 7. *Policy language model in XACML.*

Policy is an identifier of a group of rules or rule combining algorithm, it is also a group of obligation and a target. Most of the XACML processes happen in policy, the policy language model describes the basic elements and interrelationships in policy, as shown in Figure 7.

Policy contains target, rule, rule combining algorithm and obligation. The usages of these components are as follows.

Target: every policy has only one target, which help to confirm whether decision policy has correlation with request. The correlation between policy and request decides whether to evaluate the request with this policy or not, it is judged by defining the three attributes (subject, object, action) and their values in target. Compare these values in target with values of the same attributes in request, if they match, we consider policy is relevant to the request, then evaluate the request.

Rule: a policy can relevant to several rules, and every rule is combined by condition, effect and target. Condition is a statement of attributes, can be described by "True", "False" and "Indeterminate". Effect is a predict consequence of the rule, the value is "Permit" or "Deny". The target is the same as that in policy and it helps to decide the correlation between rule and request. The final result of rule is decided by the evaluation of condition. If condition returns "Indeterminate", the rule returns "Indeterminate", too. If condition returns "False", rule returns "Not Applicable". If condition returns "True", rule returns the value of effect, "Permit" or "Deny".

Rule combining algorithm: a policy has multiple rules, different rules may have conflict results. The rule combining algorithm is to solve this conflict, every policy and every request will have one final result. Every policy has only one rule combining algorithm.

Obligation: one of the main targets of XACML is to provide a fine-granted access control, and obligation is to realize this target. Obligation must be enforced together with the enforcement of authorization decision by PEP. The writer of a policy set may add obligation expressions to the policy set. After assessing the policy, PDP returns certain obligations to the PEP in its response context.

While creating policy target, the attributes and values of subject, resource and action need to be defined. When PDP is assessing the request, it will find the policy with the same attribute value in both target and request. The mechanism named "Attribute Designator" is used to compare the attribute values of request and target.

Because the OAuth cross domain identification have no way to define access control policy, it is usually combined with XACML.

3.6. Cloud Identity Federation

The cloud identity federation has become a mainly problem

as the increase of user amount. Two ways are usually used to realize cloud identity federation, one is to establish identity provider (IdP) [29] and manage users inside enterprise, another is to provide union identify management by particular provider in cloud, named identity as a service (IDaaS) [30].

The best advantage in IdP based identify federation is that this method ensure the consistency among identity, inter-enterprise policy, access management et al. The enterprise can update the existing identity management system to support identity federation instead of rebuilding a system. This method also ensures the trustworthy of the provider itself. However, it is hard for a company to manage the outside users, which means this method is not suitable in cross domain cloud environment.

In IDaaS based identity federation, cloud provider entrust identity management to particular provider. When an enterprise use IDaaS, all the identify authorization can be delivered to IDaaS, in this way, users from different domains can access the application of this enterprise, and all the users' identity information will be managed together in platform provided by IDaaS.

The mainly usage of identity federation is single sign-on(SSO) [31]. In SSO, users can sign on only once to get the authority to access systems and applications, they can switch from current environment to other business partner's environment and among several applications with no need to re-identify. The precondition of SSO is that the application system or trust domain has established an identify alliance with identify federation technology. The realization of SSO is due to the coherent and security identification and information exchange mechanism, which allows the security certificate information to deliver or share quickly in security alliance. The standard of SSO is Security Assertion Markup Language (SAML) [32], which provides a strong and scalable data format set to exchange data and identification in different environment. The identification processes of SSO model using SAML are shown in Figure 8.

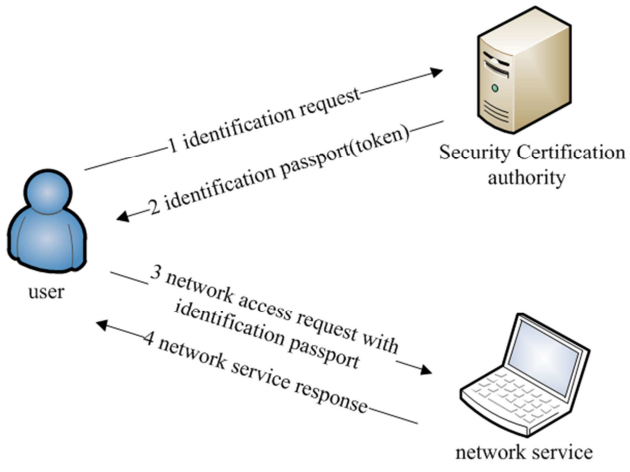

Figure 8. *Identification processes of SSO model using SAML.*

The certification authority in Figure 8 can be identity provider (IdP) or identity management service (IDaaS). Use SAML assertion as a token, the user certification in this token usually signed by certification authority. After receiving the token, network service can verify the identity of the issuer, if decode the token it will get the assertion and then identify the user who sends the request.

If the user wants to access other network, all he needs to do is to show the token signed by SAML authority to the new network service.

4. Progress of Research

4.1. Researches on ABE Based Cloud Storage Access Control

At present, many security cloud storage access control schemes based on CP-ABE have been proposed in researches, and their frameworks are similar to the one we proposed in section 2.2. Sekhar et al. [33] proposed such cloud storage architecture based on CP-ABE. In their architecture, the system is composed with four entities: key generation center, data storage center, data owner and user. The system access policy is expressed as a monotonic tree structure, only those users whose attributes set matches this policy can decrypt the message. This system is also resistant to collusion attacks in which an attacker might obtain multiple private keys. Alshehri et al. [34] applied CP-ABE algorithm in electronic health records (EHR) cloud storage to obtain security access control. They used medical institution's attributes and certificates to encrypt the information, only when user's attributes satisfy the access policy can decrypt the information. Xu et al. [35] combined CP-ABE with hierarchical key management in cloud document sharing system, so that the system can generate different secret keys for users in different security class. Thus users can preview the same document with different authorities and a fine-grained document sharing system is achieved.

In some applications, access structure may contains confidential information, thus it's necessary to do policy evaluation under cipher text form. Hidden access structure can be constructed by using hidden attributes inner-product predicates encryption (IPE) [36]. Lewko et al. [37] implemented a schema with fully hidden access structure and fully secure CP-ABE scheme based on attributes hidden IPE. In their scheme, access structure must be written in CNF (Conjunctive Normal Form) or DNF (Disjunctive Normal Form) form, so arbitrary access structures may render a super-polynomial blowup. To solve this problem, Lai et al. [38] proposed a partly hidden access structure scheme for the efficiency of CP-ABE. In this scheme, they used dual system encryption methodology to obtain fully security of system, but the scheme only supports restricted access structures expressed in AND gates on multi-valued attributes with wildcards. In Lai et al.'s another article, they proposed a partly hidden access structure scheme [39], access structure can be expressed as an LSSS, which is more flexible and expressive than previous works [17][22]. They also adopted dual system encryption methodology to obtain fully secure. In this scheme, each attribute has attribute name and attribute value, when a

user's private key attributes set do not satisfy the access structure which is in cipher text, the specific attribute values of the access structure are hidden, while other information of the access structure is public.

One shortcoming of CP-ABE is that it needs lots of computations and bandwidths to retrieve, re-encryption and re-publication large data. Some optimized models have been proposed. The simplest model to reduce the revocation consumption is called lazy revocation [40]. The main idea is to propose the whole revocation process until data are updated. But this scheme does not fit for the applications where user revocation and data update are occurred frequently, nor support security policies enforcing. To support efficient and secure revocation, some researchers [41-43] proposed proxy re-encryption schemes, for example, an optimized model combines lazy revocation with proxy re-encryption proposed by Zhang and Chen [44]. The proxy re-encryption based models transfer the re-encryption workload to third-party agents or CSPs, but the actual total consumption is not reduced and will bring in extra storage space. Another issue of proxy re-encryption is that the third-party is "trusted but curious", it will execute user's requirements but also peer into the re-encryption contents. Cheng et al. [45] introduced data splitting into their user revocation optimized model. In this model, the original data will be split into n slices via a special (n, n) threshold firstly, then data owner choose a random slice and publish it under CP-ABE algorithm, and the rest slices are published into the cloud servers directly without extra encryption. When user revocation occurred, data owner only needs to retrieve, re-encryption and republish the encrypted slice. And it does reduce the total consumption on computations and bandwidths, in spite of brought in computation on data split.

The CP-ABE based model can also be used in multi-authority access control system. Yang et al. [46] proposed DAC-MACS (Data Access Control for Multi-Authority Cloud Storage), an effective and secure data access control scheme with efficient decryption and revocation. The attribute revocation method by assigning version number for each attribute can achieve both forward and backward security: The newly joined user can decrypt the previously published cipher texts if he/she has got adequate attributes (Forward Security); The revoked user cannot decrypt the newly encrypted cipher texts which require the revoked attributes to decrypt (Backward Security). A token-based decryption outsourcing method is used to achieve efficient decryption. Li et al. [47] proposed a threshold multi-authority CP-ABE access control scheme for public cloud storage, named TMACS, the framework is similar to DAC-MACS, the main difference is: In DAC-MACS, the attribute set is divided into multiple disjoint subsets and each one of the multiple authorities maintains one attribute subset. By contrast, in TMACS, multiple authorities manage the whole attribute set together but no one has full control of any specific attribute, threshold secret sharing makes sure that no single authority can obtain the secret key. This schema satisfies the scenario of attributes coming from different authorities as well as achieves security and system-level robustness. Doyel Pal et al. [48] put forward a multilevel threshold secret sharing scheme to enhance the security of secret keys in a distributed cloud environment. At the first level the user splits the key and distributes the shares among resource providers to ensure availability. A threshold value is generated in the second level dynamically which enhances the security since the attacker cannot know the value beforehand and dummy keys are used to increases the probability of knowing if a resource provider is compromised by any attacker.

4.2. Researches on Multi-Clouds Access Control

Although ABAC model is considered as a more suitable solution for secure information sharing in multi-cloud environment than RBAC, when facing the enterprises with existing RBAC system, re-establishing an ABAC model is not realistic. Zhu et al. [49] raise a method to transfer the easy-use features of RBAC model into the ABAC model by applying ABE schemes. Firstly, establish a model named attribute-based encryption with attribute hierarchies (ABE-AH), then transforms the RBAC mechanism into an ABE-based instance. Based on this instance, data can be encrypted by using ABE and then stored into cloud. Finally define a transform policy covers transform rules, processes of encryption and decryption and key management, where the rule is a map from role to attribute.

The cross domain access control has some particular requirements. An access control policy must be flexible, expressive and able to enforce different data access permissions over the multiple groups of user from collaborative parties. User revocation cost leading to re-key generation of non-revoked users and file re-encryption must be minimized. Fugkeaw and SATO in [50], Yang and Wang in [51] presented a C-CP-ARBE model (Collaborative Ciphertext-Policy Attribute Role Based Encryption) combined RBAC with CP-ABE, which provides a more expressiveness of policy specification and less revocation cost. The policy accommodates the privilege (read or write) of users for each role distinctively. User attributes from multiple domains can be specified under the respective policy of any data owners. User decryption key graph (UDKG) is used to make all user decryption keys are securely stored in a cloud. User keys will be dynamically invoked upon the user's request for access. This provides zero cost for key distribution and enables efficient multiple keys assignment and retrieval. Secret seal (SS) is used to encrypt cipher texts to reduce the revocation cost, since SS is symmetrical triple data encryption algorithm with rapid generation process. When a user needs revocation, SS updates and re-encrypts cipher texts to get new texts.

Direct at the existing problems of flexibility, timeliness and other aspects in multi-domain access control in the current cloud, Xiong et al. [52] combined the advantage of RBAC and task driving model, and implemented a more flexible and efficient access control model. Dynamically updating role-access tables according to requests reduces the

calculation time of global strategy of synthetic, and improves the efficiency of authorization.

To solve the divulge problem of sensitive attributes in ABAC model, Peng et al. [53] presented a trust based access control in cross-domain (CD-TBAC). This model combines attributes management system with domain decision system, divides the sensitivity degree of subjects' attributes and also introduces dynamic trust metric system based on time decay. XACML access control framework is used in every single domain, attributes management system(AMS) and domain decision system(DDS) are added inter domains. AMS is used to distribute user and attribute certificate to simplify cross domain access process. DDS is to decide which domain user is when he sends an access request. The model uses trust value and attributes' value to determine the role of the subject and then determine permission by access control policy.

Some researches realized access control based on the cloud platform like Open-Stack and Amazon Web Services (AWS). Pustchi et al. [54] proposed a multi-cloud OpenStack access control (MC-OSAC), in which domain-admin is introduced to structure policy model and rule-mapping inter domains and to determine whether to trust the user.

In the ontology based access control researches, Ke et al. [15] proposed an ontology based attribute semantic matching method in cross-domain access control. The method is based on other domains' knowledge about the relationship among specific concepts, thus attributes matching problem is transferred to determining the relationship of concepts. Besides each domain needs to maintain a concept relationship database used to query concept relationship inter-domains. Zhang et al. [16] implemented an ontology based distributed access control system using OWL, where attribute mapping is realized by similarity calculating. Using OWL can clearly expresses the relationships between attributes, meets the requirements of semantic, and simplifies the attributes management in original model. Nevertheless the similarity calculating algorithm and mapping accuracy still can be optimized. Sharma and Joshi [55] presented a method using OWL to describe ABAC model. Tsai et al. [17] built a cloud access control model by using roles ontology based RBAC model, and proposed an ontology transfer algorithm for similarity calculating in different ontology. Imran-Daud et al. [56] used semantic network technology and raised a dynamic and privacy-driven access control model. By using knowledge base and language tools to replace the pre-defined sensitive words set, they propose an automatically method to assess the degree of sensitiveness. Auxilia et al. [57] put forward an ontology centric access control (OCAC) framework, which defines user, resource and action ontology, and calculate the similarity of access request on behalf of these ontology. This framework can avoid policy collision effectively and enable users to manage their policies.

In policies integration researches, Bonatti et al. [25] first proposed an algebra model for policies integration, which can make a decision of allow or deny to an access requirement, as depicted in section 3.3. Later on, Jagadeesan et al. [18] solved the history-aware access control issue by adding time restriction in this model. Backes et al. [19] defined conjunction, disjunction and scoping operators for enterprise privacy policies integration. Rao et al. [20] proposed a fine-grained policy algebra, which consists of four basic operators: '+' (addition), "&" (intersection), "¬" (negation) and "Π_{dc}" (domain projection). And a data protection model in the cloud proposed by Lin et al. [58] adopted this algebra for policies integration. Li et al. [59] raised a trust attribute-based access control algebraic system of policies composition method by introducing trust attribute, which is dynamically judged by context and time decay. This method adds trust-based vote operator and extends the authorization term to quintuple which consist of subject attribute, object attribute, environment attribute, trust attribute and operation attribute. A new access control policies composition method is proposed by Lin et al. [60] named Packet. It can detect and resolve policy conflicts in cloud service composition. The Packet method is divided into four steps. First applies a unified description to transform heterogeneous policies into a unified attribute based format. Second, improves the conflict detection efficiency by adopting cosine similarity-based algorithm. Third, exploits a hierarchical structure approach to detect policy conflicts. Finally applies conflict resolution techniques into corresponding conflict types. This method has successfully implemented in Openstack platform. Vashistha et al. [61] presented some of the most commonly used workflow heuristics currently being used in a cloud environment. Radi [62] proposed a new service broker policy for large-scale cloud applications based on the round-robin algorithm, that is implemented and evaluated using a simulator named CloudAnalyst. The author compared it with three existing policies in terms of overall average response time by using different virtual machine load balancing algorithms. Benali et al. [63] proposed an approach that is based on two essential mechanisms, context censoring and context reasoning. They considered the information acquired by the context censoring as a product line and used feature models to represent the information received, the services provided by cloud provider, the available resources and constraints.

5. Conclusion

We pointed out two issues in cloud storage access control in this paper: one is cipher text access control in single cloud; and another is cross-domain access control in multi-clouds. For the first issue, the main technology is to use CP-ABE based algorithm to implement cipher text access control in cloud storage; regard to the second, researchers usually extend ABAC model to multi-clouds access control system, including ontology based and algebra based key technologies to solve attributes mapping and policies integration issues respectively. We also introduced the main solutions in cross domain access control from three aspects: cloud identification, cloud access authorization and cloud identity federation. The solutions include OAuth cross-domain identification, XACML based access authorization, IdP and IDaaS cloud identity federation. Finally we review the progress of researches in these fields.

Based on the researches on access control in cloud storage and considering the existing problems in current researches, expecting to improve and optimize the key technologies we have reviewed in this paper in the future work, we will discuss some research directions in cloud storage access control: 1. Because of the complexity of cloud computing, most researches proposed recently are still rest on theory explore stage, there still have a lot of work to do to translate these theoretical achievements into practical applications, especially in multi-clouds access control; 2. Not only the resource data needs to be protected, but also the privacy protection of identity information when accessing cross domain is essential. So a two-way authentication protocol with privacy protecting needs to be proposed in multi clouds; 3. A cloud storage secure framework aimed at access control is still lacking, considering the heterogeneous environment of cloud, accessing control in both single cloud and multi-clouds needs to be taken into account in this framework; 4. As the rapid development of block chain technology, the cloud storage method based on block chain will be widely used in the near future. Block chain uses distributed encryption format to store data and have an ordered chain path, in which every block has the encrypted hash value of the previous block. This ensure the block chain has characteristics like transparent, cannot be tampered or denied, these characteristics can reduce the costs of trust negotiation in cross domain environment. If the resources are big, their hash value can be stored in block chain and original files in cloud storage. By combining block chain and cloud storage, the cross domain access will be simplified because of the advantages of block chain. The researches on combination of block chain and cloud storage still need to be optimized.

References

[1] CDMI TWG, "Cloud Data Management Interface (CDMI)", CDMI International Standard-2016. doi:10.3403/30334096.

[2] L. Gehlod, V. Jain, and M. Sharma, "Cloud Computing Management and Synchronization Tools", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, 2013, pp. 3026-3030.

[3] I. Livenson and E. Laure, "Towards Transparent Integration of Heterogeneous Cloud Storage Platforms", in the fourth International workshop on Data-intensive distributed computing, California, 2011, pp. 27-34.

[4] Y. D. Wang, J. H. Yang, C. Xu, X. Ling, and Y. Yang, "Survey on Access Control Technologies for Cloud Computing", Journal of Software, Vol. 26, no. 5, 2015, pp. 1129-1150.

[5] J. Crampton, "Cryptographically-enforced Hierarchical Access Control with Multiple Keys", Journal of Logic and Algebraic Programming, Vol. 78, no. 8, 2009, pp. 690-700.

[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data", in the 13th ACM conference on Computer and communications security, Virginia, 2006, pp. 89-98.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy Attribute-based Encryption", in the 2007 IEEE Symposium on Security and Privacy, Washington, 2007, pp. 321-334.

[8] Y. Wang, L. Wei, X. Tong, X. Zhao and M. Li, "CP-ABE Based Access Control for Cloud Storage", Information Technology and Intelligent Transportation Systems, Vol. 455, 2017, pp. 463-472.

[9] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage", ACM Transactions on Information and System Security, Vol. 9, no. 1, 2006, pp. 1-30.

[10] A. Sahai and B. Waters, "Fuzzy Identity-based Encryption", in the 24th annual international conference on Theory and Applications of Cryptographic Techniques, Denmark, 2005, pp. 457-473.

[11] D. F. Ferraiolo, R. S. Sandhu, S. I. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-based Access Control", ACM Transactions on Information and System Security, Vol. 4, no. 3, 2001, pp. 224-274.

[12] Y. Tian, Y. Peng, G. Gao and X. Peng, "Role-based Access Control for Body Area Networks Using Attribute-based Encryption in Cloud Storage". International Journal of Network Security, Vol. 19, no. 5, 2017, pp. 720-726.

[13] R. Dixit, S. Shivathare and G. Ganesh, "Time Domain Attribute Base Access Control for Cloud Based Content Sharing: A Cryptographic Approach", International Journal for Modern Trends in Science and Technology, Vol. 3, no. 1, 2017, pp: 74-78.

[14] E. Yuan and J. Tong, "Attributed based Access Control (ABAC) for Web Services", in the IEEE International Conference on Web Services, Washington, 2005, pp. 1-9.

[15] K. Ke, O. Li, and C. Z. Xu, "Towards Semantic Matching of Attributes in Multi-domain Access Control", in the International Symposium on Intelligence Information Processing and Trusted Computing, Washington, 2010, pp. 349-352.

[16] S. M. Zhang, H. B. Yang and B. Y. Wang, "Realization Distributed Access Control Based on Ontology and Attribute with OWL", Advances in Electronic Engineering, Communication and Management, Vol. 1, Berlin: Springer, 2012, pp. 583-588.

[17] W. T. Tsai and Q. Shao, "Role-based Access-control Using Reference Ontology in Clouds", in the Tenth International Symposium on Autonomous Decentralized Systems, Washington, 2011, pp. 121-128.

[18] R. Jagadeesan, W. Marrero, C. Pitcher and V. Sarawat, "Timed Constraint Programming: a Declarative Approach to Usage Control", in the 7th ACM SIGPLAN international conference on Principles and practice of declarative programming, New York, 2005, pp. 164-175.

[19] M. Backes, M. Dürmuth and R. Steinwandt, "An Algebra for Composing Enterprise Privacy Policies", in 9th European Symposium on Research Computer Security, France, 2004, pp. 33-52.

[20] P. Rao, D. Lin, E. Bertino, N. Li and J. Lobo, "An Algebra for Fine-grained Integration of XACML Policies", in the 14th ACM symposium on access control models and technologies, New York, 2009, pp. 63-72.

[21] L. Y. Yu, "RDFS and Ontology", A Developer's Guide to the Semantic Web, 2011, pp. 109-153.

[22] L. Y. Yu, "OWL: Web Ontology Language", A Developer's Guide to the Semantic Web, 2011, pp. 155-239.

[23] Web Ontology Language (OWL), Semantic Web Standards-2012.

[24] G. Navarro, "A Guided Tour to Approximate String Matching", ACM Computing Surveys, Vol. 33, no. 1, 2001, pp. 31-38.

[25] P. Bonatti, S. D. C. D. Vimercati, P. Samarati, "An Algebra for Composing Access Control Policies", ACM Transactions on Information and System Security, Vol. 5, no. 1, 2002, pp. 1-35.

[26] The OAuth 2.0 Authorization Framework, IETF standard-2012.

[27] N. Naik and P. Jenkins, "An Analysis of Open Standard Identity Protocols in Cloud Computing Security Paradigm", in IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 2016, pp. 428-431.

[28] EXtensible Access Control Markup Language (XACML) Version 3.0, OASIS standard-2013

[29] Identity Provider Deployment, UK Access Management Federation for Education and Research, 2008.

[30] A. Kumar, "Model Driven Security Analysis of IDaaS Protocols", in the 9th international conference on Service-Oriented Computing, Berlin, 2011, pp. 312-327.

[31] M. H. Cho, E. G. Jang, Y. R. Choi, "User Authentication Technology using Multiple SSO in the Cloud Computing Environment", Journal of The Korea Society of Computer and Information. Vol. 21, no. 4, 2016, pp. 31-38.

[32] Security Assertion Markup Language (SAML) V2.0, OASIS standard-2005.

[33] B. R. Sekhar, B. S. Kumar, L. S. Reddy and V Poornachandar, "CP-ABE Based Encryption for Secured Cloud Storage Access", International Journal of Scientific & Engineering Research, Vol. 3, no. 9, 2012, pp. 628-632.

[34] S. Alshehri, S. Radziszowski and R. K. Rajendra, "Designing a Secure Cloud-Based EHR System using Ciphertext-Policy Attribute-Based Encryption", in the Data Management in the Cloud Workshop, 2012, pp. 1-5.

[35] D. Y. Xu, F. Y. Luo, L. Gao, Z. Tang, "Fine-grained Document Sharing using Attribute-based Encryption in Cloud Servers", in the 3rd International Conference on Innovative Computing Technology, 2013, pp. 65-70.

[36] J. Katz, A. Sahai, B. Waters, "Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products", Journal of Cryptology, Vol. 26, no. 2, 2013, pp. 191-224.

[37] A. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters, "Fully Secure Functional Encryption: Attribute-based Encryption and (Hierarchical) Inner Product Encryption", in the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Berlin 2010, pp. 62-91.

[38] J. Z. Lai, R. H. Deng and Y. J. Li, "Fully Secure Cipertext-policy Hiding CP-ABE," in International Conference on Information Security Practice and Experience, 2011, pp. 24-39.

[39] J. Z. Lai, R. H. Deng and Y. J. Li, "Expressive CP-ABE with Partially Hidden Access Structures", in the 7th ACM Symposium on Information, Computer and Communications Security, New York 2012, pp. 18-29.

[40] M. Backes, C. Cachin and A. Oprea, "Lazy Revocation in Cryptographic File Systems", in the Third IEEE Int. Security in Storage Workshop, Washington, 2005, pp. 1-11.

[41] B. Libert and D. Vergnaud. "Unidirectional Chosen-ciphertext Secure Proxy Re-encryption", IEEE Transactions on Information Theory, Vol. 57, no. 3, pp. 360-379, 2011.

[42] X. H. Liang, Z. F. Cao, H. Lin and J. Shao, "Attribute based Rroxy Re-encryption with Delegating Capabilities", in the 4th International Symposium on Information, Computer, and Communications Security, New York, 2009, pp. 276-286.

[43] K. Yang, X. Jia and K. Ren, "Attribute-based Fine-grained Access Control with Efficient Revocation in Cloud Storage Systems", in the 8th ACM SIGSAC symposium on Information, computer and communications security, New York, 2013, pp. 523-528.

[44] R. Zhang and P. Chen, "A Dynamic Cryptographic Access Control Scheme in Cloud Storage Services", in 8th Int. Conf. on Computing and Networking Technology (ICCNT), 2012, pp. 50-55.

[45] Y. Cheng, Z. Y. Wang, J. Ma, J. J. Wu, S. Z. Mei and J. C. Ren. "Efficient Revocation in Ciphertext-policy Attribute-based Encryption based Cryptographic Cloud Storage", Journal of Zhejiang University SCIENCE C: Computer & Electronics, Vol. 14, no. 2, 2013, pp. 85-97.

[46] K. Yang, X. Jia, K. Ren, B. Zhang and R. T. Xie. "DAC-MACS: Effective Data Access Control for Multi-authority Cloud Storage Systems", IEEE Trans. Inf. Forensics Security, Vol. 8, no. 11, 2013, pp. 1790–1801.

[47] W. Li, K. P. Xue, Y. Xue and J. Hong, "TMACS: A Robust and Verifiable Threshold Multi-Authority Access Control System in Public Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, Vol. 27, no. 5, 2016, pp. 1484-1496.

[48] D. Pal, P. Khethavath, J. P. Thomas and T. chen, "Multilevel Threshold Secret Sharing in Distributed Cloud", in Security in Computing and Communications. 2015, pp. 13-23.

[49] Y. Zhu, D. J. Huang, C. J. Hu and X. Wang, "From RBAC to ABAC: Constructing Flexible Data Access Control for Cloud Storage Services", IEEE Transactions on Services Computing, Vol. 8, no. 4, 2015, pp. 601-616.

[50] S. Fugkeaw and H. Sato, "An Extended CP-ABE based Access Control Model for Data Outsourced in the Cloud", in the IEEE 39th Annual Computer Software and Applications Conference, Washington, Vol. 3, 2015, pp. 73-78.

[51] X. H. Yang and H. Wang, "A Cross-Domain Access Control Model Based on Trust Measurement", Wuhan University Journal of Natural Sciences. Vol. 21, no. 1, 2016, pp. 21-28.

[52] D. Xiong, P. Zou, J. Cai and J. He, "A Dynamic Multi-domain Access ControlModel in Cloud Computing", International Symposium on Security in Computing and Communication, Vol. 536, 2015, pp. 3-12.

[53] W. P. Peng, X. Z. Liu, H. R. Guo and C. Song, "Research on Trust Based Access Control in Cross Domain", Application Research of Computers, Vol. 33, no. 6, 2016, pp. 1790-1796.

[54] N. Pustchi, F. Patwa and R. Sandhu, "Multi Cloud IaaS with Domain Trust in OpenStack", in the Sixth ACM Conference on Data and Application Security and Privacy, New York, 2016, pp. 121-123.

[55] N. K. Sharma and A. Joshi, "Representing Attribute Based Access Control Policies in OWL", in IEEE Tenth International Conference on Semantic Computing, 2016, pp. 333-336.

[56] M. Imran-Daud, D. Sánchez and A. Viejo, "Privacy-driven access control in social lnetworks by means of automatic semantic annotation", Computer Communications, Vol. 76, 2016, pp. 12-25.

[57] M. Auxilia and K. Raja, "Ontology Centric Access Control Mechanism for Enabling Data Protection in Cloud", Indian Journal of Science and Technology. Vol. 9, no. 23, 2016, pp. 1-7.

[58] D. Lin and A. Squicciarini, "Data Protection Models for Service Provisioning in the Cloud", in the 15th ACM symposium on access control models and technologies, New York, 2010, pp. 183-192.

[59] Y. Y. Li, H. R. Guo, W. P. Peng and C. Song. "Trust Attribute-based Access Control Policies Composition", Application Research of Computers. Vol. 33, no. 7, 2016, pp. 2175-2180.

[60] L. Lin, J. Hu and J. Zhang. "Packet: a Privacy-aware Access Control Policy Composition Method for Services Composition in Cloud Environments", Frontiers of Computer Science, Vol. 10, no. 6, 2016, pp. 1142-1157.

[61] A. Vashistha, R. Porwal, A. K. Soni. "A Taxonomy of Scheduling Algorithms for Cloud Computing", International Journal of Computer Science Issues, Vol. 12, no. 1, 2015, pp. 67-71.

[62] M. Radi. "Efficient Service Broker Policy For Large-Scale Cloud Environments", International Journal of Computer Science Issues, Vol. 12, no. 1, 2015, pp. 85-90.

[63] A. Benali, B. E. Asri, H. Kriouile. "Toward Sensor and Software Product Line Based Context Aware Cloud Environment Assignment", International Journal of Computer Science Issues, Vol. 13, no. 5, 2016, pp. 76-85.