JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 743-758 (2017)
DOI: 10.6688/JISE.2017.33.3.9

A PANH-based Access Control Mechanism for Cross-Cloud Service Composition*

AO-DI LIU, NA WANG AND MING-CONG LIU
National Digital Switching System Engineering and Technological Research Center
ZhengZhou Science and Technology Institute
Zhengzhou, 450000 P.R. China
E-mail: {ladyexue; tinatwf}@163.com; lmc340406@sina.com

For access control problem of cross-cloud service composition, we propose an access control mechanism of cross-cloud service composition. The mechanism uses policy attribute negotiation based on historical information (PANH) to achieve access control. The mechanism can ensure the consistency of the different service components policies in global composite service and avoids that the composite service does not work properly because of policy conflicts. We have designed a negotiation algorithm based on historical information. Historical information is used in negotiation process. By synchronizing high frequency negotiation policy, storing history information of negotiation and calculating cost of attributes disclosure, we optimize the negotiation process and improve the efficiency of negotiation. Finally, our simulations verify the feasibility and efficiency of the mechanism.

Keywords: cloud service, service composition, access control, ABAC, policy negotiation

1. INTRODUCTION

As the main application form [1] of cloud computing, cloud service [2] has obtained a very wide range of applications. Single cloud service can only provide limited capabilities. By implementing the integration of different cloud services, cloud composite [3] service can meet different application requirements flexibly and effectively. However, lots of new security requirements are attendant as well. The process of formation of cloud composite service needs integration of such services which come from different security management domain, so it is almost impossible to achieve consistent access control by simply superimposing access control policy. Meanwhile, the interactive process of access among cross-cloud service components has such features: dynamic, flexible, open, no-central, *etc.* In addition, at the time of access control for cross-cloud service resource, minimizing the disclosure of sensitive information can reduce the risk of security information leakage. The features above mentioned present new challenges for access control mechanisms.

For the access control issue of cross-cloud service composition, based on RBAC model, PAN [4] uses role permissions relation tree to present a visual role mapping algorithm and achieves the integration of cloud access control policy, whereas it cannot achieve the interaction of the cross-cloud service access control. Literatures [5, 6] propose a distributed access control framework for cloud computing. Access control man-

Received July 16, 2016; revised September 28, 2016; accepted October 22, 2016.
Communicated by Ram Chakka.
* This research was supported by the National High Technology Research and Development Program (863 Program) of China (NO. 2015AA011705), and Natural Science Foundation of Henan Province of China (No. 162300410334).

agement in the clouds through using role mapping mechanism can achieve mutual access among different security domains, but this mechanism requires a unified trusted third party to support. In dynamic cloud environments, it's easy to cause performance bottlenecks. Meanwhile, if the third-party occurs security threats, it will have a marked impact on all cloud services. Satoh [7] divides security policies of composite service and proposes composite rule for security policy. But the premise is that we need a consistent policy logical expression among the different service components, which is difficult to be achieved in realistic complex cloud environment. Boella [8, 9] proposes a method based on game theory in knowledge sharing, according to the policy knowledge of different service components, do a game calculation to get the priority of different policy enforcement. Lin [10] is based on ABAC model, Srivatsa [11] is based on historical information of service call and Bruns [12] is based on the four-valued logic of Belnap bilattice, they perform access control for composite service by policies composition algebra. But the composite process need to disclose all access control policies, which improves the safety risks. By embedding the access control policies in the service composition algorithm, Chou [13] achieve the unity of service availability and safety. However, after composite service being changed, the structure of access control needs to be adjusted, which is not suitable for dynamic cloud environment.

In order to solve the problems encountered by current cross-cloud service composition, this paper achieves access control of cross-cloud service composition based on the theory of attribute negotiation. It should be noted that some scholars have proposed access control scheme based on trust negotiation, for example these literatures [14, 15] propose a negotiation model based on Colored Petri Net (CPN) in cloud environment, but it does not consider the composite relationship among service components. Ma [16] proposes a scheme of automated trust negotiation based on fuzzy logic. The negotiation path can be optimized with more concise and flexible description by modeling the credential access control policies in fuzzy logic formula. But it can only be used for the attribute of having a value constraint, so that the range of application is limited. Li [17] utilizes DFA (deterministic finite automaton) to describe the negotiation process, which improves reasoning capability of negotiation. Squicciarini [18] utilizes historical information in the resource negotiation process, while there is insufficient to utilize historical data that is limited to the frequent request for the same resource. Further, different from traditional automated trust negotiation that only consider both negotiation parties, there are various composition relationships among different service components in cloud composite service. The same resource may be controlled by multiple service components, and conflicts may arise among different policies of component. The integration of all service components access control policies for the same resource is necessary, so that access control situation is more complex and diverse. The result of negotiation needs to be able to be transferred among different service components and we call it as the trust transfer. So we achieved access control for cloud composite service by trust transfer and multi-party negotiation.

This paper is based on ABAC (attribute based access control) model [19, 20]. This model can meet the dynamic, fine-grained access control requirements and is compatible with other access control model (such as DAC, MAC, RBAC, *etc.*). On the basis of in-depth analysis for composite service relationship, we design an access control mechanism for cross-cloud service composition based on policy attribute negotiation of histor-

ical information. The mechanism achieves multi-party distributed negotiation among cross-cloud service components by transferring the result of negotiation to maintain a dynamic chain of trust. The no-central and distributed negotiation process is able to adapt to the dynamic, flexible feature of cloud environment. And it avoids disclosure of policy privacy information. In addition using historical negotiation information optimizes negotiation algorithm and improves the efficiency of negotiation.

The remainder of the paper is organized as follows: Section 2 introduces the policy attribute negotiation model. In Section 3, we present a policy attribute negotiation algorithm based on historical information. In Section 4, we provide detailed experimental results to verify the availability and effectiveness of the mechanism. Section 5 concludes the paper.

2. POLICY ATTRIBUTE NEGOTIATION MODEL

2.1 Term and Symbol Definition

Definition 1: Attribute, it is the basic unit of describing physical characteristics, recorded as *ATTR* and divided into public attribute, locked attribute, private attribute. Public attribute is not limited and disclosed directly, recorded as *ATTR_T*. Locked attribute requires attribute negotiation. When the result of negotiation is ok, the attribute will be disclosed, recorded as *ATTR_L*. Private attribute is confidential and cannot be disclosed, recorded as *ATTR_P*.

Definition 2: Attribute set, it is a set to describe entity's attributes, recorded as *ATTR_SET*. $ATTR\_SET = \{A|ATTR\_T \cup ATTR\_L \cup ATTR\_P\}$.

Definition 3: Access request, it is a request that service accesses resource, recorded as $request \rightarrow \{R, ATTR\_SET, sign\}$. *R* represents the requested resource, *sign* is the response result of request.

Definition 4: Access control policy, it is a set of relevant rules that subject accesses object and represents an action of authorization. The policy specifies which attributes need to be owned by entity to access appropriate resources, and is represented by *POLICY*: $R \leftarrow \Theta\{ATTR\_SET\}$. *ATTR_SET* represents an access control attribute set, Θ represents the logical relationships of conjunction or disjunction. When the request meets logical expression $\Theta\{ATTR\_SET\}$, *R* can be accessed.

Definition 5: Negotiation policy, it specifies conditions for the disclosure of attribute and is the rules that control policy attribute negotiation, recorded as *NEO_POLICY*: $ATTR \leftarrow \Theta\{ATTR\_SET\}$. When $\Theta\{ATTR\_SET\}$ is true, the attribute *ATTR* is disclosed.

Negotiation policy is the core of interactive negotiation and determines which attributes will be disclosed by negotiating parties. It needs to meet the following requirements: (1) negotiation policy must be comprehensive. If a negotiation is successful in theory, it will be successful in fact; (2) Negotiation can be suspended; (3) It needs to minimize the disclosure of unrelated attributes; (4) It is feasible and the cost of commu-

nication and computation is in a reasonable range.

Definition 6: Unlocking operation, recorded as *UNLOCK* ($ATTR\_L_I$, $ATTR\_SET_J|R_J$), for the request that entity *I* want to access entity *J*'s resource *R*, use negotiation policy to reason the negotiation result. If it is ok, attribute R_J will be unlocked. Then the attribute will be disclosed.

Definition 7: Policy attribute negotiation, based ABAC model, interacts to disclose attributes according to access control policy and negotiation policy, and builds trust relationship between access requester and resource provider. The negotiation of entity *I* and entity *J* can be expressed by, $N = \{request, ATTR\_SET_I, ATTR\_SET_J, POLICY_I, POLICY_J, NEO\_POLICY_I, NEO\_POLICY_J, NEO\_TAG, Nsign\}$. *NEO_TAG* is negotiation mark and records service component information of multi-party negotiation process for trust transfer. *Nsign* represents the negotiation result, success or failure.

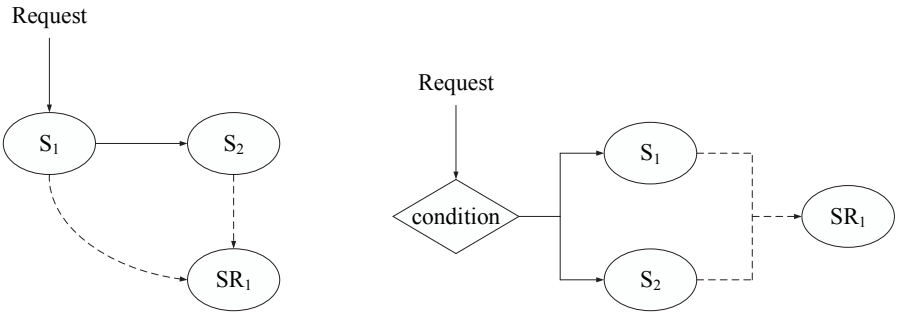
2.2 Features of Cross-cloud Service Composition Negotiation

In cloud composite service, it is not a simple point to point negotiation relationship for two service components, and may involve multiple service components. There is a transfer of trust relationships. The initial access control policy of the service component is generally established according to their internal security requirements. In the process of composite service interaction, the policy decision of service request may be performed by a plurality of service components. Different service components may produce a different decision result based on local access control policy. Meanwhile, in order to protect the security and privacy of each service component, we cannot disclose their all access control policies to maintain the consistency of composite service policy. In this paper, the issue of consistency can be addressed by multi-party negotiation among service components. Parties involved in the negotiation need to achieve multi-party's trust transfer by maintaining a chain of trust.

Specific to the cross-cloud service composition problem, this paper analyzes consistency problems that timing control (TC) relationship and selection control (SC) relationship of compose service may arise. For TC relationship, when the access request transfer exits and different components decisions of the same request are inconsistent, it will result in conflict. For SC relationship, when selection conditions are inconsistent with the request to access resource, it will result in conflict, too. To sum up, mainly the following two forms negotiation: (1) other cloud service components directly access to resources; (2) due to the different service components storing access control policies for the same resource, trust needs to be transferred.

From a global perspective view of composite service, resource request interaction of service components is an inside interaction. But from the local perspective view, it is an external interaction. As shown in Fig. 1 (a), from the local to the analysis, the component S1 receives a request for resource SR1 from the top component. S1 needs to obtain operating authority of SR1 through S2. This time needs negotiation between components S1 and S2, the result is if top request is able to operate SR1. This time may produce two types of negotiation. When S1 does not exist access control policy for SR1 and S2 exits access control policy for SR1, SR1's permission management will be controlled solely

by S2, S1 is used to forward the top resource request. However when S1 and S2 exit policies for SR1 at the same time, the response result of top request will be influenced by both S1 and S2. Because S1 and S2 belong to different service component, conflicts possibly appear. In SC relationship of service component, service conditions and S1, S2 may appear conflicts as shown in Fig. 1 (b).



(a) TC relationship (b) SC relationship

Fig. 1. The service composite relationship.

Conflict resolution: when conflict occurs, to simplify the process, each service component only needs to be responsible for themselves' access control module. For the request transferring to the local, if the negotiation result is deny request, request is rejected directly. If the result is allow request, we will store the result to *NEO_TAG*. *NEO_TAG* will be transferred to next service component according to composite relationship structure for transfer of trust. Then the next service component with its local access control agent determines whether to allow access until the process of this request is completed. The process flow is shown in Fig. 2.

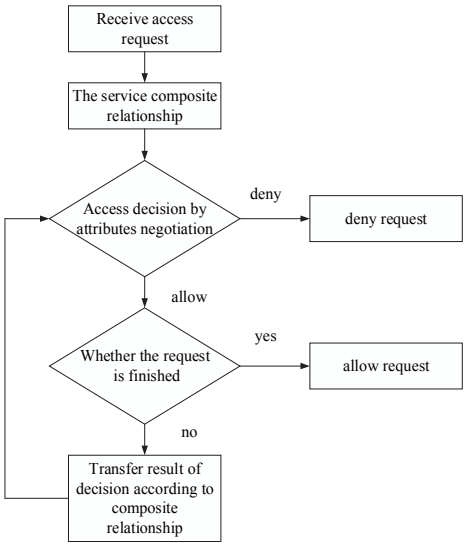


Fig. 2. Multi-party negotiation process.

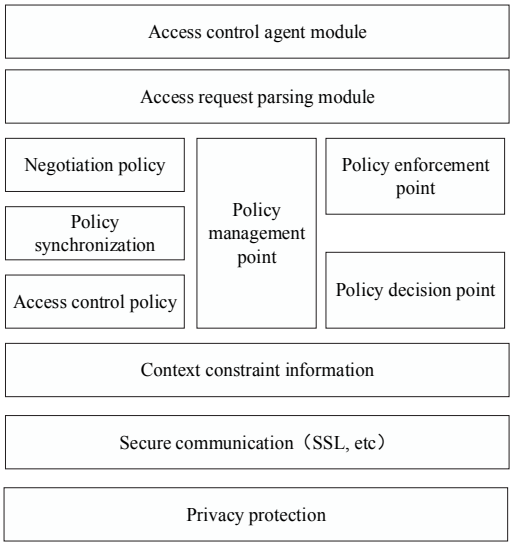


Fig. 3. Access control agent module.

2.3 Multi-party Negotiation Mechanism

Each service component negotiates policy attributes by access control agent. As shown in Fig. 3, access control agent module includes: access request parsing module, policy enforcement point, policy management point, negotiation policy, policy synchronization, access control policy, policy decision point, context constraint information, secure communication, *etc.* Access control agent receives resource request from other cloud service. The resource is protected by the local cloud's access control policy. Agent needs to analyze attribute information from resource requester and make access control decision.

The following is a detailed procedural description. First, request parsing module parses attribute information of request and determine whether the access request is transferred by other service component, what type of negotiation is required, which is the key to maintain a chain of trust among the service components. Negotiation policy, access control policy and policy synchronization module are managed uniformly by policy management module which provides policy support for policy enforcement point and policy decision point. At the same time, access control is constrained by context constraint information. Communication Security of negotiation interaction builds on SSL encryption method. The entire access control agent provides technical support for the privacy and security of service components. After the attributes of request are negotiated to meet the authorization requirements, requesters can make the allowed access and operation for resource. By multi-party attribute negotiation mechanism among access control agents, we achieve fine-grained access control for cross-cloud service composition without exposing service sensitive information.

2.4 The Example of Negotiation

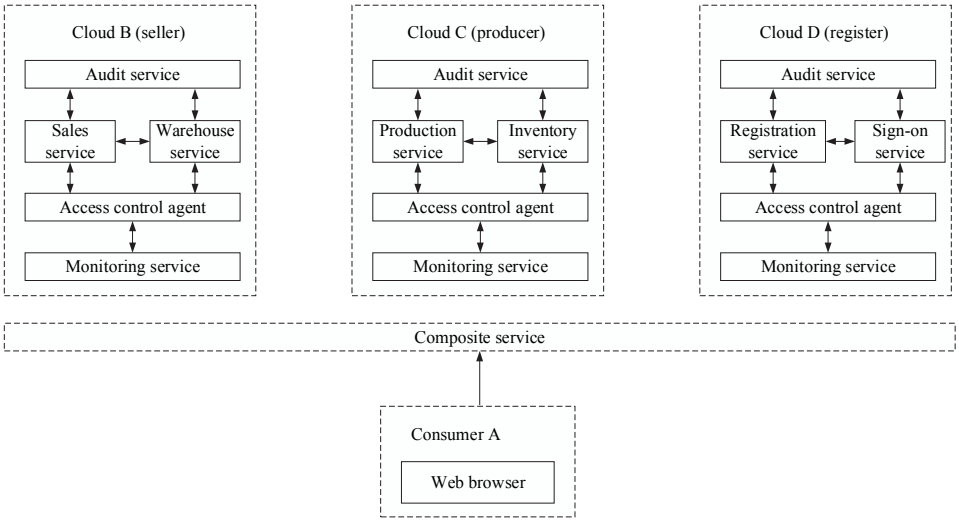
In the background of cross-cloud service composition, Fig. 4 (a) shows an application example of the multi-party negotiation interaction. The example includes service components from three cloud organizations. They are cloud B as a seller, cloud C as a producer, and cloud D as a register. There are two major internal service components which are sales service and warehouse service in the cloud B. There are two major internal service components which are production service and inventory service in the cloud C. There are two major internal service components which are registration service and sign-on service in the cloud D. The product d is a resource to be processed in composite service. Fig. 4 (b) constitutes a graph of service composite relationship for Fig. 4 (a). Consumer A proposes a purchase request of d. Composite service checks whether consumer A is a legitimate user by sign-on service and has the proof of purchase by sales service. If A does not have the proof, it will need to register by registration service and then purchase. After successful login, consumer A and sales service establish customer relation. 'A' will delivery from warehouse service. If the quantity of good is zero in the warehouse, warehouse will need to call inventory service to delivery. Production service needs to verify the attribute information of the seller, when the decision of access control agent is allowed, inventory service will provide product d for sales service. If inventory service does not have d, it will need to use production service to produce d. Finally, composite service will return the response result of purchasing d. In the entire process of

cloud composite service, d is provided by composite service for consumer A. Seller sells product, producer manufacture product, and register is responsible for user registration. Different service components in the cloud control resource at the same time and have their own access control policy for the resource. PolicyB is B's access control policy for d, PolicyC is C's access control policy for d, and PolicyD is D's access control policy for d. In the access control process for consumer request, seller, producer and register by access control agent resolve the request. Combined with negotiation policy of the parties, we will maintain the chain of trust by disclosure of relevant attributes and get the negotiation result. Eqs. (1)-(3) describe the negotiation process.

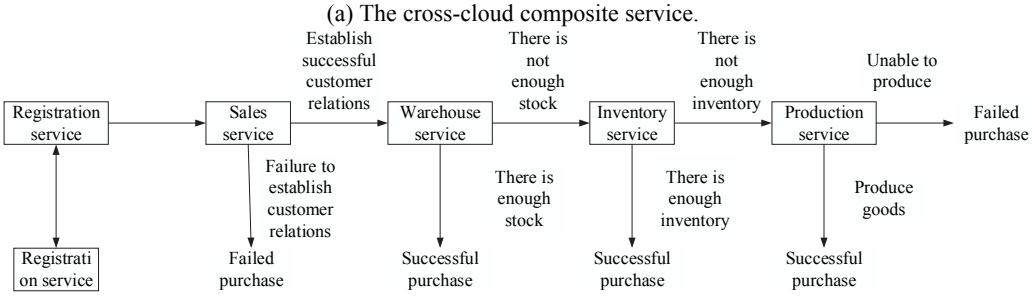
$$N_{AB} = \{request, ATTR\_SET_A, ATTR\_SET_D, POLICY_A, POLICY_D, NEO\_POLICY_A, NEO\_POLICY_D, NEO\_TAG, Nsign_{AD}\}, \quad (1)$$

$$N_{DB} = \{request, ATTR\_SET_D, ATTR\_SET_B, POLICY_D, POLICY_B, NEO\_POLICY_D, NEO\_PDOLICY_B, NEO\_TAG_{DB}, Nsign_{DB}\}, \quad (2)$$

$$N_{BC} = \{request, ATTR\_SET_B, ATTR\_SET_C, POLICY_B, POLICY_C, NEO\_POLICY_B, NEO\_PDOLICY_C, NEO\_TAG_{BC}, Nsign_{BC}\}. \quad (3)$$



(a) The cross-cloud composite service.



(b) The graph of service composite relationship.

Fig. 4. The example of negotiation.

3. NEGOTIATION ALGORITHM BASED ON HISTORICAL INFORMATION CORRECTION

3.1 The Method of Negotiation

Multi-party negotiation will be converted to the two sides of negotiation of point to point by transferring negotiation mark *NEO_TAG* to maintain a chain of trust in different service components. The main mode of negotiation policy disclosure has full disclosure and interactive disclosure in the negotiation process. Full disclosure means that before starting negotiation, the two sides will disclose all of the negotiation policy. The advantage is that an interaction will finish the negotiation process. It is high efficiency. However, it is necessary to expose all negotiation policy that is easy to generate security threats. Interactive disclosure means that participants deduce attributes that can be safely unlocked according to the unlocked attribute and negotiation policy. The advantage is that it is not necessary to disclose own negotiation policy. The disadvantage is that interactive communication overhead is too large.

Negotiation mode has positive negotiation and reverse negotiation. Positive negotiation: when requester sends out disclosure request of an attribute, receiver resolves request and send the needed attribute set. Requester checks whether the attribute disclosure condition is satisfied. If there are some not unlocked attributes, the disclosure request of these attributes will be sent until all the attribute have been unlocked or some attributes must not be unlocked. Finally, suspend negotiation and return to the result of response. Benefits: only disclose attribute which access control policy and negotiation policy involve. Irrelevant information will not be disclosed and service's privacy information can be better protected. Disadvantages: negotiate step by step following the hierarchy. Negotiation rounds is more.

Reverse negotiation: when requester sends out disclosure request of an attribute, receiver resolves request and send all public attributes. Requester computes to get all attributes that can be disclosed by combining with negotiation policy and send them to receiver. The receiver computes whether access control policies are met and send a response. Benefits: four interaction will accomplish the negotiation. The cost is relatively low. Disadvantages: all the attributes will be disclosed and privacy can't be protected, there are security risks.

In the paper, we adopt partial disclosure mode and positive negotiation. Partial disclosure combines the advantages of full disclosure and interactive disclosure. It is the balance for privacy protection and negotiation efficiency. Negotiation process is divided into two stages that are pre-negotiation stage and formal negotiation stage. In the pre-negotiation stage, according to historical records of negotiation, the used high frequency negotiation policy will be synchronized to the negotiating party. The use of pre-negotiated reasoning can come to additional unlocked attributes. Then negotiation will begin. This approach reduces policy's disclosure and improves the performance. In the formal negotiation stage, based on historical negotiation information, agent will disclose attribute by steps according to privacy disclosure weight value. Relationships of negotiation stage and policies are shown in Fig. 5.

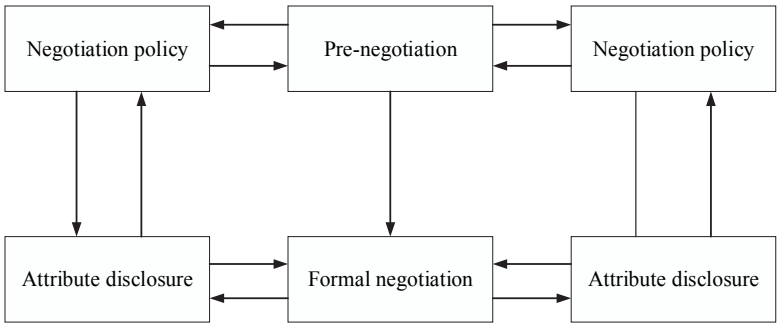


Fig. 5. The relationships of negotiation stage and policies.

3.2 The Method of Weight Calculation

From the three aspects, negotiation based on historical information has been amended.

(1) Negotiation policy's disclosure mode: according usage frequency of negotiation policy, the policy of high frequency will be disclosed at the pre-negotiation stage;
(2) Selection of attributes negotiation path: combining the possibility of attribute's disclosure and the weight of disclosed cost, we compute the weight value of negotiation entrance path for target resource. The path of low weight value will be negotiated preferentially for reducing the cost of privacy disclosure and improving the success rate of negotiation.
(3) The successfully negotiated set of attributes: we will every attribute that has been successfully negotiated place in historical records of service component by the form of attribute label. The label has a valid period. Within the valid period, the attribute of label need not to be repeated negotiated.

The current negotiation mechanism is mainly researched in the method of negotiation and ignore the usage of history information. Historical information not only includes the historical results of negotiation, but also reflects the correlation among attributes. Usage of historical information will increase negotiation efficiency. The negotiation weight value is evaluated based on historical information. It can reduce the round of negotiation and cost of attribute disclosure to improve the effectiveness and safety of negotiation.

The cost of attribute privacy disclosure is calculated by negotiation policy that discloses the attribute. Negotiation policy can been expressed in the form of a disjunctive paradigm and decomposed into a plurality of conjunctive expression, as long as a conjunctive expression is met that the policy will be met. Then the attribute will be disclosed. Combining the cost of attribute privacy disclosure, we get Eq. (4),

$$WEIGHT(n) = c\sum_{t\in Bn}\frac{1}{Nt}\sum_{m\in t}WEIGHT(m) + (1-c)I(n). \tag{4}$$

$WEIGHT(n)$ represents the cost of attribute n, Bn is a set of conjunctive expressions

that n's negotiation policy is decomposed into. Set t is the involved set of attributes in conjunctive expression. Value m is the specific attribute in t. Nt is the number of set t. Value c is an adjustment parameter ($0 < c < 1$). $I(n)$ is the initial weight of attribute n.

When the negotiation process encounters multiple path to choose, we assume that the disjunctive expression set of current negotiation path is Sn. $E\_OLD(m)$ is a historic expectation of successful negotiation for attribute m. E_NEW represents the result of attribute negotiation that is the value of $Nsign$. Success is 1, failure is −1. $E\_ATTR(m)$ represents the successfully negotiated expectation of attribute m. $WEIGHT(m)$ is the disclosure cost of attribute m. α ($0 < \alpha < 1$) and β are adjustment parameters. E_ROUTE is the selected negotiation path. Eqs. (5) and (6) are shown below.

$$E\ ATTR(m) = (1 - \alpha) \times E\quad OLD(m) + \alpha \times E\quad NEW, \tag{5}$$

$$E\_ROUTE = \underset{t \in Sn}{MAX}\{\sum_{m \in t}(E\_ATTR - \beta \times WEIGHT(m))\}. \tag{6}$$

The α is more closing to 0, then this negotiation will have a smaller influence for selecting negotiation path and sample has the greater impact. Update is slow. The α is more closing to 1, then this negotiation will have a greater influence for selecting negotiation path and sample has the smaller impact. Update is fast. During the negotiation, whenever we encounter the problem of choosing path, it will be calculated by the above method.

3.3 The Example of Interactive Negotiation Process

Examples of policy set as shown in Table 1:

Table1. Negotiation policy set.

policy set of service component I	policy set of service component J
R	$R \leftarrow (J_1 \vee J_2) \vee (J_3 \vee J_5) \vee J_6$
$I_1 \leftarrow J_5 \vee J_6$	$J_1 \leftarrow P$
$I_2 \leftarrow J_4$	$J_2 \leftarrow I_2 \vee I_3$
$I_3 \leftarrow T$	$J_3 \leftarrow I_1 \vee I_4$
$I_4 \leftarrow J_2$	$J_4 \leftarrow T$
$I_5 \leftarrow T$	$J_5 \leftarrow T$
$I_6 \leftarrow J_2 \vee J_4$	$J_6 \leftarrow I_5 \vee I_6$

In the process of attributes negotiation between the two sides, exposed attributes sequence can be represented by multi-tree, called attribute negotiation tree (ANT) that consists of attribute nodes and disjunctive nodes. As shown in Fig. 6, round node represents attribute node, square node represents disjunctive node. Fig. 6 simulates negotiation process of Table 1 for resources. Fig. 6 shows that there are multiple paths to be negotiated successfully. From node R, we begin to negotiate attributes by traversing ANT. T node represents public attribute, P node represents private attribute. Private attribute can't be disclosed, so its cost is ∞. Negotiation mechanism based on historical information can improve efficiency and reduce disclosure of attribute privacy information.

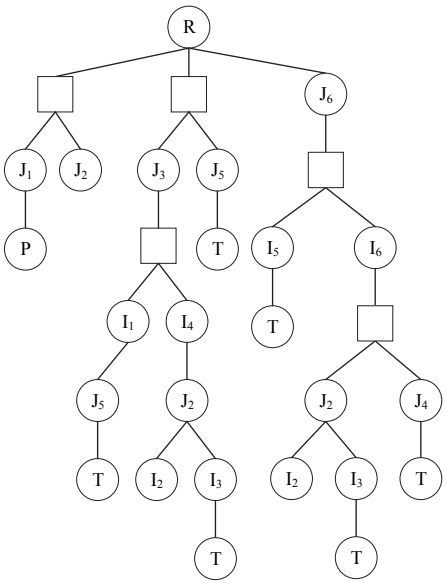


Fig. 6. The attribute negotiation tree (ANT).

Negotiation interactive diagram of Table 1 is shown below.

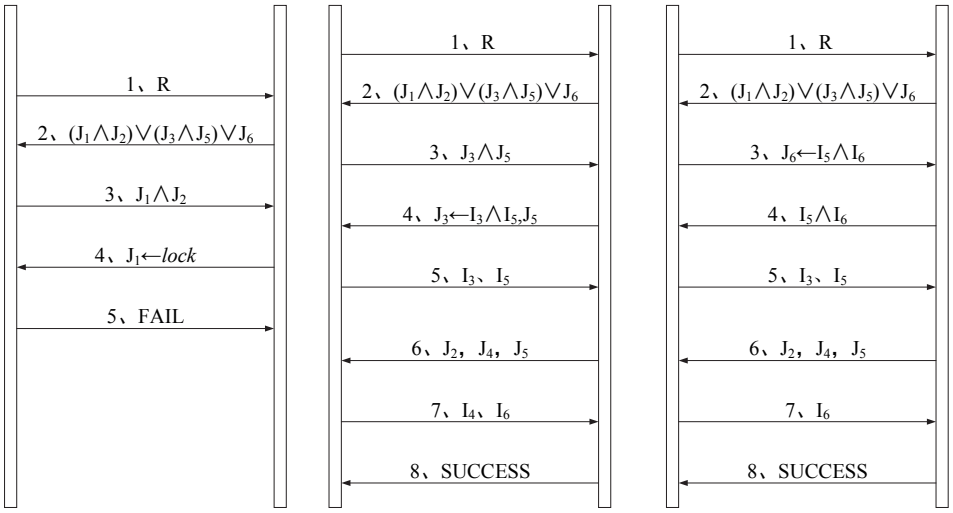


Fig. 7. The negotiation interactive process.

3.4 Negotiation Algorithm

Negotiation is achieved by the disclosure of attributes. The actual negotiation process can be abstracted as ANT shown in Fig. 6. However, from the point of negotiator, every negotiator can't get the entire ANT, service component's security private information is well protected. Negotiation algorithm based on historical information can use

historical information to choose negotiation path, maintain a set of attributes that has been successfully negotiated and reduce the round of negotiation.

Table 2. Negotiation algorithm based on historical information.

Input: ResourceRequest, NegotiationPolicySet, PolicySet
Output: the result of negotiation

```
Msg = Receive_msg();
ExpPolSet = GetPolicyExp(); //Get entry of access policy attributes.
Switch(Mst.type)
{
    case NegotiationInfo:
            //Determine whether the attribute has been negotiated and whether the
              attribute is public attribute.
            if (NegNodeSet out of HistoryPulishNode and AttNode is null)
                send NegotiationSuccessInfo
            //Determine whether there are private attributes.
            else if (AttNodeSet is null && NegNodeSet include lock_node)
                send ChangeRouteInfo
            else
            {
                Reduce(AttNodeT, AttNodeSet)
                Add(NewUnlockSet, AttNodeT)
                ExpNegSet = GetNegotiationExp() //Get entry of negotiation policy
                attributes
            //Get optimal entry of negotiation policy attributes according to historical
              information and privacy weight.
                ExpBetSet = GetBetterRoute(ExpNegSet)
                Send (NewUnlockSet, ExpBetSet)
            }
    case ChangeRouteInfo: //Change negotiation path.
            ExpBetSet = GetOtherBetterRoute()
            Send (NegotiationInfo, ExpBetSet)
    case NegotiationSuccessInfo:
            negotiation success
            negotiation end
    default:
            negotiation failure
            negotiation end
            }
```

4. EXPERIMENT AND ANALYSIS

In order to verify feasibility and effectiveness of the mechanism, this paper is based on the attribute set and policy set of standard policy compliance test package provided by XACML and based on extended platform of TrustBuilder. We simulate a cross-cloud

service composition scene of Section 2.4 to build simulation environment. We design and implement three different experiments to analyze the feasibility and effectiveness of mechanism. Meanwhile, we analyze the influence of negotiation historical scale for efficiency. Since the purpose of the proposed method uses policy attribute negotiation to achieve the normalization of cross-cloud service composition function and completes interaction of resource information. Therefore, in a specific experiment, we determine the feasibility and effectiveness of the method by negotiation time. In addition, we determine whether the method can achieve efficient policy negotiation by comparing time of positive negotiation, reverse negotiation and negotiation based on historical information under the premise of privacy attribute protection. Finally, experiment tests the influence of parameters α and β for negotiation efficiency.

In experiment 1, we compare the required negotiation time of the traditional positive negotiation, traditional reverse negotiation, policies composition algebra, and negotiation based on historical information. In order to prove the reliability and rationality of experiment, we select 5 groups of attribute sets which contain a different number of private attributes. Their corresponding numerals are 1-5, the scale of attribute sets are 100, 200, 500, 1000, 2000. Respectively, we use the same negotiation policy set to do 200 negotiation experiments and record an average rate of successful negotiation and time cost. The final results are shown in Fig. 8.

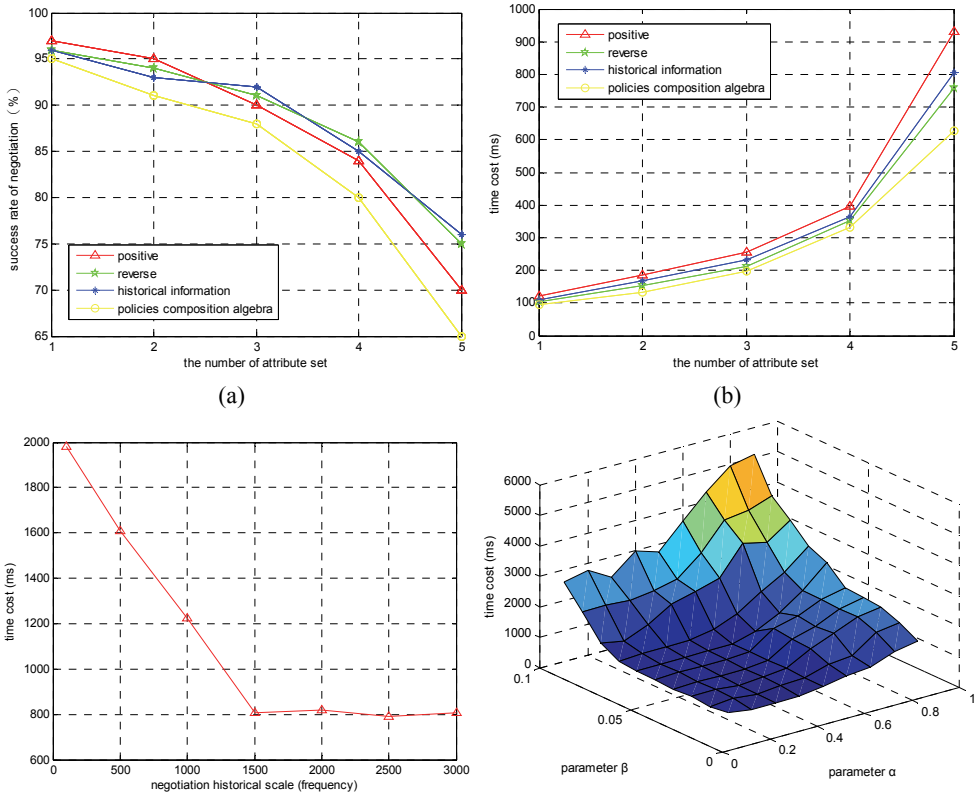


Fig. 8. The results of experiments.

In experiment 2, in order to verify the influence of history information for negotiation efficiency, we tested the time cost of different historical negotiation's scales.

In experiment 3, we verify the influence of parameters α and β for negotiation efficiency. Then, in the range of parameters, we test time cost of completing the negotiation by changing the size of parameters.

Figs. 8 (a) and (b) show that at low scale of attributes, the four kinds of negotiation schemes all have a high success rate of negotiation. But with the growth of attribute scale, negotiation success rate will decline. Success rate of policies composition algebra declines most significantly. Success rate of negotiation based on historical information is lower than positive negotiation and reverse negotiation at low scale of attributes. But with the growth of attribute scale, it can reach approximate success rate of reverse negotiation. Since reverse negotiation will disclose all of the attribute information of negotiator and policies composition algebra will disclose all of the access control policy, negotiation based on historical information has a stronger ability of privacy protection under the premise to ensure the success rate of negotiation.

Fig. 8 (c) shows that time cost decreases significantly with the improvement of negotiation historical scale. After reaching the scale about 1500, the time will be stabilized. For a particular negotiation process, negotiation historical information can only optimize the process and reduce the number of negotiation round, but it can't affect the success or failure of the negotiation. When the scale is sufficient and the negotiation process has been optimized, the negotiation also needs the minimum time cost and the influence of historical scale will be limited, which is consistent with the experiment result in Fig. 8 (c). In the Fig. 8 (d), we find that parameters α and β have a significant impact on the efficiency of Negotiation. Parameter α controls the update frequency of history information, parameter β controls the influence of disclosure weight for choosing the negotiation path. When $\alpha > 0.3$, $\beta < 0.06$, the time cost of negotiation is maximum. With the decrease of α, the time cost is minimum in $\alpha = 0.3$. However, with the decrease of β, time cost also increases. When $\beta < 0.06$, time cost will stabilize. The β is higher, the percentage of disclosure weight is higher. Then the privacy protection effect is better. $\beta = 0.06$ can play a good influence. Therefore, $\alpha = 0.3$, $\beta = 0.06$ are the optimal values.

According to the three experimental results, we can get conclusion. Negotiation based on historical information can be applied to the scene of cross-cloud service composition. It is feasible and effective.

5. CONCLUSIONS

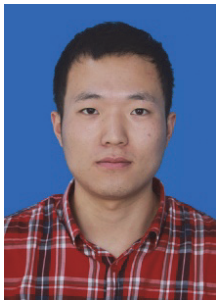
The paper presents an access control mechanism based on policy attribute negotiation of historical information for cross-cloud service composition. We identify the goals and general process of negotiation. Subsequently the safety protection of private information in different cloud has been achieved. Negotiation algorithm based on historical information optimizes the negotiation process and improve the efficiency of negotiation. The mechanism can meet the needs of access control in cloud environment and provides protection for the security of cross-cloud service composition. On account of the research of this paper mainly focuses on the access control problem of cross-cloud service composition, we will study the access control problem of the intra-cloud composite ser-

vice in the next step. Moreover, it will be a meaningful work to propose an integrated solution to integrate the access control mechanism of the cross-cloud and intra-cloud composite service.

REFERENCES

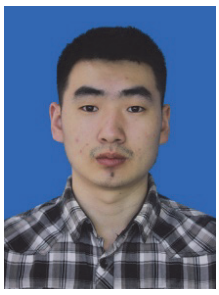
1. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, Vol. 53, 2010, pp. 50-58.
2. M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, "Cloud computing: Distributed internet computing for IT and scientific research," *IEEE Internet Computing*, Vol. 13, 2009, pp. 10-13.
3. A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Systems with Applications*, Vol. 41, 2014, pp. 3809-3824.
4. L. Pan, N. Liu, and X. Zi, "Visualization framework for inter-domain access control policy integration," *Wireless Communication Over Zigbee for Automotive Inclination Measurement China Communications*, Vol. 10, 2013, pp. 67-75.
5. A. Almutairi, M. Sarfraz, S. Basalamah, W. Aref, and A. Ghafoor, "A distributed access control architecture for cloud computing," *IEEE Software*, Vol. 29, 2012, pp. 36-44.
6. B. Li, M. Tian, Y. Zhang, and L. Shenjuan, "Strategy of domain and cross-domain access control based on trust in cloud computing environment," *Lecture Notes in Electrical Engineering*, Vol. 277, 2014, pp. 791-798.
7. F. Satoh and T. Tokuda, "Security policy composition for composite web services," *IEEE Transactions on Services Computing*, Vol. 4, 2010, pp. 314-327.
8. G. Boella and L. van D. Torre, "Security policies for sharing knowledge in virtual communities," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part A: Systems and Humans*, Vol. 36, 2006, pp. 439-450.
9. G. Boella and D. T. L. Van, "A game theoretic approach to contracts in multiagent systems," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part C: Applications and Reviews*, Vol. 36, 2015, pp. 68-79.
10. L. Lin, J. P. Huai, and X. X. Li, "Attribute-based access control policies composition algebra," *Journal of Software*, Vol. 29, 2009, pp. 403-414.
11. M. Srivatsa, A. Iyengar, T. Mikalsen, I. Rouvellou, and J. Yin, "An access control system for web service compositions," in *Proceedings of IEEE International Conference on Web Services*, 2007, pp. 1-8.
12. G. Bruns, D. S. Dantas, and M. Huth, "A simple and expressive semantic framework for policy composition in access control," in *Proceedings of ACM Workshop on Formal Methods in Security Engineering*, 2007, pp. 12-21.
13. S. C. Chou and J. Y. Jhu, "Access control policy embedded composition algorithm for web services", in *Proceedings of IEEE International Conference on Advanced Information Management and Service*, 2010, pp. 54-59.
14. H. Chen, Q. Chen, and C. Wang, "A CPN-based trust negotiation model on service level agreement in cloud environment," *International Journal of Grid and Distrib-*

uted Computing, Vol. 8, 2015, pp. 247-258.
15. C. Wang, Q. Chen, H. Chen, and H. Xu, "An SLA-oriented multiparty trust negotiation model based on HCPN in cloud environment," *International Journal of u-and e-Service*, *Science and Technology*, Vol. 8, 2015, pp. 321-336.
16. X. X. Ma and G. S. Zeng, "Scheme of automated trust negotiation based on fuzzy logic," *Computer Science*, Vol. 42, 2015, pp. 220-223.
17. H. Lu and B. Liu, "DFANS: A highly efficient strategy for automated trust negotiation," *Computers and Security*, Vol. 28, 2009, pp. 557-565.
18. A. Squicciarini, E. Bertino, E. Ferrari, and F. Paci, "PP-trust-X: A system for privacy preserving trust negotiations," *ACM Transactions on Information and System Security*, Vol. 10, 2007, p. 12.
19. X. Jin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering DAC, MAC and RBAC," in *Proceedings of IFIP Annual Conference on Data and Applications Security and Privacy*, 2012, pp. 41-55.
20. X. M. Wang, F. U. Hong, and L. C. Zhang, "Research progress on attribute-based access control," *Acta Electronica Sinica*, Vol. 38, 2010, pp. 1660-1667.



Ao-Di Liu (刘敖迪) received his B.S. degree from Zheng-Zhou Science and Technology Institute, Zhengzhou, China, in 2014. He is currently working toward an M.S. degree at Zheng-Zhou Science and Technology Institute. His research interests include cloud computing and network security.

Na Wang (王娜) received her B.S., M.S., and Ph.D. degrees from ZhengZhou Science and Technology Institute, Zhengzhou, China, in 2001, 2004, and 2008, respectively. She is currently an Associate Professor of Zhengzhou Science and Technology Institute. Her research interests include cloud computing and trust management.



Ming-Cong Liu (刘明聪) received his B.S. degree from ZhengZhou Science and Technology Institute, Zhengzhou, China, in 2015. He is currently working toward an M.S. degree at Zheng-Zhou Science and Technology Institute. His research interests include cloud computing and service management.