Access Control Policy Analysis and Access Denial Method for Cloud Services

Chi-Lun Liu

Abstract—Personal cloud computing is an emerging trend in the computer industry. For a sustainable service, cloud computing services must control user access. The essential business characteristics of cloud computing are payment status and service level agreement. This work proposes a novel access control method for personal cloud service business. The proposed method sets metadata, policy analysis rules, and access denying rules. Metadata define the structure of access control policies and user requirements for cloud services. The policy analysis rules are used to compare conflicts and redundancies between access control policies. The access denying rules apply policies for inhibiting inappropriate access. The ontology is a theoretical foundation of this method. In this work, ontologies for payment status, access permission, service level, and the cloud provide semantic information needed to execute rules. A scenario of personal data backup cloud service is also provided in this work. This work potentially provides cloud service providers with a convenient method of controlling user access according to changeable business and marketing strategies.

Index Terms—Access control, cloud services, ontology, payment, service level agreement.

1. Introduction

Cloud computing is an important trend in the Internet and computer technologies. In the emerging cloud computing model, servers can be dynamically provisioned and configured to deliver services in a scalable manner[1]. In this model, personal applications and data are kept not only on the personal computers of users, but also in the cloud operated by a cloud computing service provider. Since cloud computing service providers rarely know their users in advance, assigning user roles in access control policies is difficult[2]. Therefore, developing a new access control mechanism for the new context of cloud computing is an important theoretical and practical issue.

The business use of cloud computing has two essential characteristics: the payment status and service level agreement. The payment status can be defined as an accounting record of one stage in the payment process used by cloud providers. In practice, cloud computing access control models usually consider service accounting records of cloud computing providers[2]. Cloud computing service providers often use a pay-per-use strategy to charge service fees. In contrast, charges in the pay-per-use strategy are based on the quantity used[3]. The above discussion indicates that the payment status for service fees is important for cloud access control.

A service level agreement (SLA) is a service contract that defines the level of service and controls the use of computing resources[2]. Cloud computing service providers generally use an SLA to guarantee the service level that the enterprise has committed to provide[3]. An SLA should also be considered in cloud computing access control models[2].

To address cloud computing business characteristics and access control requirements, this work proposes an access control method based on the payment status and service level agreement for the personal data storage. The proposed access control method comprises policy and requirement metadata, policy analysis rules, and access denial rules. The proposed metadata reveal what information should be considered in the access control policy and specify the user requirements. For example, a policy metadata comprises four elements: the payment status, access permission, service level agreement, and cloud computing service. An example of an access control policy is: the paid (payment status) user can access (access permission) a 10 G (service level agreement) personal backup cloud service (cloud computing service). The proposed policy analysis rules reveal redundant or conflicting policies. The proposed access denial rules also inhibit illegal access requests.

The theoretical foundation of the proposed access control method is the use of ontologies, which is the shared and explicit conceptualization of a domain knowledge structure[4], to provide semantic information needed to execute policy analysis and access denying rules.

The rest of this paper is structured as follows. Section 2 discusses related works in access control. Section 3 introduces the proposed access control method and its use.
of metadata and rules. A scenario in Section 4 is used as an example of a personal backup service in cloud computing. Section 5 presents the evaluation of the proposed method. Section 6 discusses the implication of the proposed method from a managerial perspective. The final conclusion section summarizes the contribution of this study.

2. Related Work

Table 1 summarizes related works in the literature. Le et al. proposed an access control method for a hospital ubiquitous information system, in which access is authorized according to user activities. Masoumzadeh and Joshi used ontologies and policy rules to control access in online social networks. Hu et al. used a semantic web technology to control cloud computing access. Zhu et al. extended role-based access control (RBAC) to the cloud computing context. Ruj et al. provided an attribute-based access control mechanism in the data storage cloud service.

Most of the above access control methods focus on cloud computing. However, few of the methods in these works address the relationships between the payment issue and access control mechanism. Few works consider the service level issue as an important element in the access control methods. In contrast, although these methods usually apply ontologies, few existing methods discuss the policy conflict issue. Therefore, this work proposes an ontological access control method based on the payment status and service level agreement to address the above issues.

3. Proposed Method

Fig. 1 shows how the proposed access control method is used in a situation involving four actors and six steps, where the actors are the end user, administrator, access control server, and cloud. The six steps of the usage process of the access control method are as follows. In the first step, the end user requests the access control server to provide a specific cloud computing service. The access control server then applies policies for confirming whether the end user has the authority to use the cloud computing service according to the payment status and service level request in step 2. The policies are managed by the administrator. After confirming that the end user has the authority, the access control server performs steps 3 and 4 to request and acquire the required cloud resources. If the end user has no authority, steps 3 and 4 are skipped, and the access control server records service logs. The final step is service delivery to the end user.

The proposed metadata have four core elements: the payment status, access permission, service level, and cloud. The metadata (Fig. 2) comprise two cloud policies and one user requirement. The proposed policy analysis rules (Table 2) are used to analyze the four relationships \( R_{PS2,PS2}, R_{AP1,AP2}, R_{SL1,SL2}, \) and \( R_{C1,C2} \) between access control policies \( ACP_1 \) and \( ACP_2 \) to identify conflicts and redundancies in the two policies. The other four relationships among the control policy \( ACP \) and user requirement \( UR \) are used by the proposed access denial rules (in Table 3) to deny and allow access. In Table 2 and Table 3, “\( \lor \)” denotes “or”, Eq is “equal”, Ex is “exclusion”, \( K \) is “kind”, and \( A \) is “antonym”.

Fig. 1. Access control method overview.

Fig. 2. Metadata of paired access control policies and a user requirement.
Table 2: Policy analysis rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>RPS1:PS2</th>
<th>RAP1:AP2</th>
<th>RSL1:SL2</th>
<th>RC1:CL2</th>
<th>Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>RulePA1</td>
<td>$\lor$ K $\land$ A</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>Policy conflict</td>
</tr>
<tr>
<td>RulePA2</td>
<td>Ex $\land$ K</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>Policy conflict</td>
</tr>
<tr>
<td>RulePA3</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>Policy redundancy</td>
</tr>
</tbody>
</table>

Table 2 summarizes the three proposed policy analysis rules for detecting conflicts and redundancies. These rules, which are developed according to the proposed metadata in Fig. 2, are described as follows.

RulePA1: IF, an equality or kind relationship exists between payment status PS1 and PS2; AND IF, an antonym relationship exists between access permission AP1 and AP2; AND IF, an equality or kind relationship exists between service levels SL1 and SL2; AND IF, an equality or kind relationship exists between clouds C1 and C2; and THEN, access control policies ACP1 and ACP2 are in conflict.

RulePA2: IF, an exclusion relationship exists between payment status PS1 and PS2; AND IF, an equality or kind relationship exists between access permission AP1 and AP2; AND IF, an equality or kind relationship exists between service levels SL1 and SL2; AND IF, an equality or kind relationship exists between clouds C1 and C2; and THEN, access control policies ACP1 and ACP2 are in conflict.

RulePA3: IF, an equality or kind relationship exists between payment status PS1 and PS2; AND IF, an equality or kind relationship exists between access permission AP1 and AP2; AND IF, an equality or kind relationship exists between service levels SL1 and SL2; AND IF, an equality or kind relationship exists between clouds C1 and C2; and THEN, access control policies ACP1 and ACP2 are in conflict.

Three rules based on the metadata in Fig. 2 are proposed to deny illegal access. These access denial rules are introduced as follows and summarized in Table 3.

RuleADIO1: IF, a quality or kind relationship exists between payment status PS1 and PS2; AND IF, an access action AP3 is an antonym of access permission AP1; AND IF, the required service level RSL3 equals or is a kind of service level SL3; AND IF, cloud C3 is equal to or a kind of cloud C1; and THEN, the user access is denied because the access control policy ACP1 inhibits the user requirement UR3.

RuleADIO2: IF, an exclusion relationship exists between payment status PS1 and PS2; AND IF, the access action AP3 is equal to or a kind of access permission AP1; AND IF, the required service level RSL3 equals or is a kind of service level SL3; AND IF, cloud C3 is equal to or a kind of cloud C1; and THEN, the user access is denied because the access control policy ACP1 excludes the user requirement UR3.

RuleADIO3: IF, an equality or kind relationship exists between payment status PS1 and PS2; AND IF, the access action AP3 is equal to or a kind of access permission AP1; AND IF, the required service level RSL3 exceeds service level SL3; AND IF, the cloud C3 is equal to or a kind of cloud C1; and THEN, the user access is denied because the user requirement UR3 exceeds the service level limitation of access control policy ACP1.

4. Cloud Service Data Backup Scenario

The scenario used as an example in this section is a personal data backup service for cloud computing. This scenario involves four ontologies and six examples for each rule. These ontologies, which include payment status ontology, access permission and action ontology, service level ontology, and data backup service ontology, are depicted in Fig. 3 to Fig. 6.

Table 3: Access denial rules

<table>
<thead>
<tr>
<th>RuleADIO</th>
<th>RPS1:PS2</th>
<th>RAP1:AP2</th>
<th>RSL1:SL2</th>
<th>RC1:CL2</th>
<th>Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleADIO1</td>
<td>$\lor$ K $\land$ A</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>Access denied because the policy inhibits the user access</td>
</tr>
<tr>
<td>RuleADIO2</td>
<td>Ex $\land$ K</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>Access denied because the policy excludes the user access</td>
</tr>
<tr>
<td>RuleADIO3</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ RSL3 $\lor$ SL3</td>
<td>$\lor$ K $\land$ Eq</td>
<td>$\lor$ K $\land$ Eq</td>
<td>Access denied because the user requirement exceeds the agreed service level</td>
</tr>
</tbody>
</table>

Fig. 3. Payment status ontology.

Fig. 4. Access permission and action ontology.

Fig. 5. Service level ontology.

Fig. 6. Data backup service ontology.
The above ontologies (Fig. 3 to Fig. 6) and the metadata (Fig. 2) are applied in the following example of a data backup service to demonstrate and validate the proposed rules in Table 2 and Table 3. The example of RulePA1 to RulePA3 demonstrates the proposed policy analysis rules given in Table 2. In the RulePA1 example, ACP1 is “the unpaid (PS1) user cannot access (AP1) any service level (SL1) personal data backup service (C1)”. And ACP2 is “the unpaid (PS2) user can access (AP2) a 1 GB (SL2) personal data backup service (C2)”. Here, PS1 equals PS2. Fig. 4 indicates that AP1 (can not access) is an antonym of AP2 (can access). Fig. 5 reveals that SL2 (1 GB) can be any SL1 (i.e., any service level). The C1 equals C2. According to RulePA1 in Table 2, ACP1 and ACP2 are in conflict. The example of RulePA2 is a conflict between ACP1 and ACP3. The ACP1 is “only the paid (PS1) user can access (AP1) 10 GB (SL1) personal data backup service (C1)”. The ACP2 is “the paid VIP (PS2) user can access (AP2) 10 GB (SL1) personal data cloud service (C2)”. Fig. 3 shows that, in this case, PS1 (only paid) excludes PS2 (unpaid). The AP1 equals AP2. SL1 equals SL2. The C1 equals C2. According to RulePA2 in Table 2, ACP1 and ACP3 are in conflict. The RulePA3 example is a redundancy between two policies. The ACP1 is “only the paid (PS1) user can access (AP1) a 10 GB (SL1) personal data cloud service (C1)”. The ACP2 is “the paid (PS2) user can upload data to (AP2) a 10 GB (SL2) personal data backup service (C2)”. Here, Fig. 3 indicates that a kind relationship exists between PS1 and PS2 and between AP1 and AP2. Both SL1 and SL2 are 10 GB. The C1 equals C2. According to RulePA3 in Table 2, ACP1 and ACP2 are redundant policies. The examples of RuleAD1 to RuleAD3 demonstrate the proposed access denial rules shown in Table 3. The RuleAD1 example shows why access by a user Jason (in Fig. 6) is denied by the policy. In this case, ACP1 is “the unpaid (PS3) user cannot access (AP1) any service level (SL1) personal data backup service (C1)”. The UR3 is “the unpaid (PS3) user Jason download data from (AP3) 10 GB (RSL3) Jason’s data backup service (C3)”. The UR3 means “data downloaded by an unpaid user Jason from a 10 GB data backup service”. In the above situation, PS1 equals PS2. Fig. 4 shows that AP1 is an antonym of AP3. Fig. 5 reveals that RSL3 is a kind of SL1. Fig. 6 shows C3 is a kind of C1. According to RuleAD1, the access is denied because the policy ACP1 inhibits the UR3 access. The RuleAD2 example is a case of access exclusion. In this example, ACP1 is “only the paid (PS1) user can access (AP1) a 10 GB (SL1) personal data backup service (C1)”. The UR3 is “a 30-day free trial (PS3) user, Alice, uploads data to (AP3) 10 GB (RSL3) Alice’s data backup service (C3)”. The UR3 means “30-day free trial for a user Alice to upload 10 GB of data to the data backup service”. Fig. 3 shows that PS3 excludes PS1. Fig. 4 reveals that AP3 is a kind of AP1. The SL1 equals RSL3. Fig. 6 presents that C3 is a kind of C1. According to RuleAD2, the user access is denied because the policy ACP1 excludes UR3 access. In the example of RuleAD3, the user requirement exceeds the agreed service level. The ACP1 is “only the paid (PS1) user can access (AP1) 10 GB (SL1) personal data backup service (C1)”. The UR3 is “a paid (PS3) user John uploads data to (AP3) 15 GB (RSL3) John’s data backup service (C3)”. The PS3 (only paid) is a kind of PS1 (paid) in Fig. 3. The AP3 is a kind of AP1 in Fig. 4. RSL3 exceeds SL1. Additionally, C3 is a kind of C1. According to RuleAD3, the access is denied due to the service level excess problem.

5. Method Evaluation

The “service level” and “cloud” are two important classifications of cloud objects. The proposed method classifies objects according to the service “level” and “cloud”. The first formula calculates the object concept number of the proposed method:

\[
\text{Num(Proposed\_Method\_Object)} = \sum_{i=0}^{k} \text{Num(Cloud}(i)\text{)} + \text{Num(Service\_Level}(i)\text{)}. \tag{1}
\]

The RBAC mechanism only uses “object” to represent the controlled resource. The RBAC mechanism does not consider the “service level”. A cloud might have several service levels, and a set of service levels might be applied in a set of clouds. The second formula for calculating object concept number in RBAC is

\[
\text{Num(RBAC\_Object)} = \sum_{i=0}^{k} \text{Num(Cloud}(i)\text{)} \times \text{Num(Service\_Level}(i)\text{)}. \tag{2}
\]

The following example shows how (1) is applied. In group 1, four service levels are used to describe three clouds. In group 2, two service levels are used to describe another cloud. The three clouds in group 1 are the data backup service, personal computer (PC) data backup service, and mobile device data backup service. The four service levels in group 1 are any service level, 1 GB, 10 GB, and 15 GB. The cloud in group 2 is a game service. Two service levels for the game service in group 2 are the general version and professional version. According to (1), the object concept number of the proposed method is

\[
\text{Num(Proposed\_Method\_Object)} = (3+4)+(1+2) = 10.
\]

If RBAC is applied in the above example, group 1 needs twelve object concepts and group 2 needs two object concepts. The twelve object concepts in group 1 are “any service level-data backup service”, “1 GB-data backup
service”, “10 GB-data backup service”, “15 GB-data backup service”, “any service level-PC data backup service”, “1 GB-PC data backup service”, “10 GB-PC data backup service”, “15 GB-PC data backup service”, “any service level-mobile device data backup service”, “1 GB-mobile device data backup service”, “10 GB-mobile device data backup service”, “15 GB-mobile device data backup service”. The two object concepts in group 2 are “general version-game service” and “professional version-game service”. According to (2), the object concept number in RBAC is

\[
\text{Num}(\text{RBAC\_Object}) = 3 \times 4 + 1 \times 2 = 14.
\]

According to the above two formulas and their examples, the proposed method obtains fewer cloud objects compared with RBAC both in term of \(\text{Num}(\text{Cloud}(i))\) and \(\text{Num}(\text{Service\_Level}(i))\) ≥ 2.

6. Managerial Implications

The above discussion has three managerial implications for the information systems and marketing departments of cloud service enterprises:

1) Assigning a specific service level for each customer type is convenient because the proposed access method classifies customers according to the service level and payment status. Cloud service providers can use the proposed method to maintain access control policies used to assign service levels to customers according to payment status.

2) Access control policies are intuitively aligned with marketing strategies because the proposed method uses payment status and service level based policies to control access. Cloud services are often marketed at different prices and service levels. Therefore, the proposed method intuitively applies the optimal marketing strategy for cloud service providers.

3) The method alleviates policy conflicts and redundancies because it provides rules for detecting conflicts and redundancies between access control policies. Therefore, it can decrease the cost of managing policy conflicts and redundancy problems.

7. Conclusions

The novel approach of the proposed algorithm is based on the ontology theory from both billing and contract perspectives. One theoretical contribution of the method is the application of ontologies to new phenomena. This work proposes an innovative theoretical set of rules for policy analysis and access denial. The practical contribution of the proposed method is to support billing and contract matters, which are essential elements in the cloud computing business model. The proposed method can facilitate cloud service providers in controlling user access according the account receivable, received payment, and agreed service level.

References


Chi-Lun Liu was born in Taiwan in 1976. He received the Ph.D. degree in management information systems from National ChengChi University (NCCU), Taipei, in 2008. He is currently an associate professor with the Department of Multimedia and Mobile Commerce, Kainan University, Tanshui. His research interests include electronic commerce security, software engineering, internet marketing, and social websites.