A Composite Access Control Model Based on Attribute and Role

Jing-yu WANG¹,a*, Yu-liang GUAN¹,b and Wei-zhen LUO²,c

¹School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou, Inner Mongolia Autonomous Region, China
²School of Software Engineering, Xiamen University, Xiamen, Fujian Province, China
abtu_wjy@qq.com, bshow.happy@hotmail.com, cwilliam@stu.xmu.edu.cn
*Corresponding author

Keywords: Access Control, Privacy Protection, Attribute, Role.

Abstract. Aimed at the problem of user privacy protection in cloud computing environment, this paper proposes a composite access control model based on attribute and role. The method is a combination of the traditional single access control model. The technology assigns roles based on attribute strategy first and subsequently assigns permissions based on roles, to make the subject operate object by access control. The simulation experimental results show that compared with the traditional single access control model, this model can effectively reduce the leakage of private data under the premise of the efficiency guarantee.

Introduction

The recent years saw the rapid development and popularity of the cloud computing technology. However, it brings prominent security and privacy issues. After the cloud users upload data to the cloud server, the privacy of their data can no longer be guaranteed by cloud service provider. Users take the risk of privacy disclosure. Access control is an effective measure which can make no effort protecting user’s privacy.

Role-Based Access Control (RBAC) has become the most widely used access control mechanisms for research and application. It introduces the concept of roles between users and access permissions. The model associates users with roles, and controls users’ access to resources in the system by authorizing roles.

Attribute-Based Access Control (ABAC) describes the authorization and access control constraints using the unified modeling of subject, object, operation and environment attributes. The authorization of access permission depends on whether the attribute provided by the user passes the system-provided attribute discriminant.

Nevertheless, references show that the traditional single access control model is not qualified for the current complex cloud environment. It is difficult to realize fine-grained access control strategy in RBAC model for it lacks a specific description of the subject and object information. What is the deadliest is that it cannot do a good job against the joint attacks. The ABAC model is lack of efficient encryption and decryption mechanism, effective policy conflict detection mechanism and active authorization mode. Not bringing in the concept of role, a large number of user's authorization management becomes extremely complicated and tedious.
Composite Access Control Model

Formal Definition of the Model

In order to distinguish between RBAC and ABAC, the Composite Access Control (CAC) model is used to name the access control model presented in this paper. The CAC model in the cloud environment provides some improvements to traditional RBAC and ABAC model. Similar to the ABAC model, the CAC model can adapt to large-scale dynamic access; also, like the RBAC model, flexible authorization and assignment are the two main characteristics.

In determining the overall framework of the model, the paper gives the definition of CAC model as follows:

The CAC model consists of the following five sets of entities: Subject (hereinafter abbreviated as S), Object (O), Environment (E), Role (R) and Permission (P).

In the access control policy of the CAC-ABAC phase, the authorization decision is made by the attributes of the entities involved in the decision. At this stage, I use Sattr, Oattr, and Eattr to represent the attribute name of subject, object and environment respectively. The consequence of assigning an attribute name is denoted as Attribute Value Pair: avp←(attr=value). Where attr is the attribute name and value is the attribute value. Savp, Oavp and Eavp are the representatives of the subject, object and environment attribute name value pair respectively.

Attribute-based Access Request (AAR) is formalized defined as follows. The model extracts the attribute values from the natural access request (NAR) and formalizes it as AAR (s, o, e). If x approaches to infinity, s={Savp 1, Savp2, …, Savpx} is the set of subject attribute name value pairs; o={Oavp 1, Oavp2, …, Oavpx} is the set of object attribute name value pairs; e={Eavp 1, Eavp2, …, Eavpx} is the set of environment attribute name value pairs.

AAR (s, o, e)={Savp 1, Savp2, …, Savpx}∩{Oavp 1, Oavp2, …, Oavpx}∩{Eavp 1, Eavp2, …, Eavpx} is derived from the above definition.

Attribute Predicate Value Pair (APVP) can be represented by apvp←(attr∝value).∝∋{>, <, ≥, ≤, ≠, in, not in, between} is a relational expression operator, which is used to define the range of values for an attribute. Savp, Oavp and Eavp are used to express the attribute predicate value pairs of subject, object and environment respectively. When y approaches positive infinity, several definitions are shown as follows: APVPS={Savp 1, Savp2, …, Savpy} is the set of subject attribute predicate value pairs; APVPo={Oavp 1, Oavp2, …, Oavpy} is the set of object attribute predicate value pairs; APVPe={Eavp 1, Eavp2, …, Eavpy} is the set of environment attribute predicate value pairs.

A CAC-ABAC policy consists of Target, Algorithm, and Rule. The policy can be represented by Policy←(Target, Algorithm, Rule). Target is composed of S, O, and E to describe whether the relevant policy is appropriate for the access request. It can be represented by Target←(S, O, E). The model uses these three elements to match the content of the request. If the match is successful, then the required policy can apply the request. The formula for Rule is Rule=Sign←(AVPV s, APVPo, APVPe). Sign is the result of the rule decision. It is defined as Sign∋(permit, deny). The Rule set is Rule={rule1, rule2, …, rulez}, where z approaches positive infinity.

In the CAC-RBAC phase, S→R represents a one-to-one relationship between subjects and roles. R→P represents a many-to-many relationship between roles and permissions. The model first assigns the role based on the subject attributes. If the subject attribute sa satisfies an attribute role expression, a role r will be assigned to the
subject s. Next, if a permissions p contains the object attribute oa, then the permission p will be assigned to the role r.

**Implementation Process of Access Control**

According to the formal definition of CAC shown above, this model combines a common strategy description language XACML. A composite access control model diagram is as shown in Fig. 1.

**Attribute-Based Access Control Phase (CAC-ABAC)**

The CAC-ABAC phase has the following modules: Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Administration Point (PAP) and Attribute Authority (AA). The stage has two requests as follows: Natural Access Request (NAR) and Attribute-based Access Request (AAR).

In the CAC-ABAC phase, the subject is authenticated first. When some subject accesses, the PEP receives the NAR from the user, and then converts the natural access request to the AAR based on the subject-related attributes and attribute values provided by the AA. The attributes and attribute values are then brought into the PDP to determine if they conform to one or more of these rules. If they do, the subject would have the right to access and get a corresponding role; otherwise, the request of the subject would be rejected by PDP. The rules above come from the PAP, and are expressed as regular expressions.

The Policy Evaluation and Decision Algorithm details the implementation of the CAC-ABAC policy evaluation and decision. For a given access request AAR (s, o, e) and access control policy Policy(Target, Algorithm, Rule), the phase first conducts a strategic assessment, PolicyEvaluation (AAR, Policy). The PolicyEvaluation judges whether the access control policy is appropriate for determining this access request. If this policy is not appropriate to determine this request, the system returns to a result as “not-applicable”; otherwise, it returns to the policy’s Sign set.
Role-Based Access Control Phase (CAC-RBAC)

For the CAC-RBAC module itself, the communication from the subject to the object is a virtual communication. As shown in Fig. 2, there is a Role Service. For the subject, all of its requests are sent to the Role Service. After that, the Role Service and the permission would figure out a result, and then the result would be sent to the subject by the Role Service.

The CAC-RBAC phase is divided into two steps. First, the roles are assigned based on subject attributes. Second, permissions are assigned based on roles. Specifically, if the attribute of the subject conforms to the regular expression of the CAC-ABAC phase, it has the right to access the object. The system assigns it a corresponding role and lets it enter the CAC-RBAC phase. Otherwise, the user's request is denied. Then, the permission binds the role. The role gets corresponding permissions as long as the subject get the role. At this point, the subject can perform the operation to access the object.

CAC-RBAC implementation processes are as follows:

Step1: The system assigns the role Roles_i to the subject S_i according to the subject attribute Sattr and the subject role authorization rule.

Step2: The system analyzes the role Roles_i owned by S_i, and grants the operation permission P_i to S_i according to the role authorization rules. This permission is for the object O_i in the cloud computing environment.

Step3: The system returns the object O_i to the subject S_i.

![Figure 2. CAC-RBAC Structure.](image-url)

The composite design of attribute and role is the key of this article. The subject attribute is the core that determines the role assignment. In the traditional RBAC model, there are many-to-many relationships between users and roles, and hence the problem of role conflict occurs. For the current RBAC technology, if a user gains more than one role, there may be conflict. Therefore, several relevant improvements are proposed in the CAC model. This paper takes a more space-consuming and time-saving approach. Specifically, each subject passing the CAC-ABAC stage of authentication will have an attribute expression. More roles are designed to ensure that each attribute expression (representing a class of subjects that can perform the same operation) is assigned a unique role to obtain all the rights that can be obtained. That is, subjects and roles are one-to-one or many-to-one relationships. In this way, there is no conflict in the distribution of roles.

To achieve this, the priority of role (or permission) must be defined in the model (the priority of the permission can also represent the priority of the corresponding role). This ensures that the relationship between the attribute expression and the role is
one-to-one. The specific method is that even if a number of attributes of a subject in line with a number of rules, it only gets a role eventually.

Simulation Experiment

In order to explain the process better and verify the correctness, this paper implements and tests the proposed scheme. In this paper, the experimental environment includes the Intel Core i7-4720HQ 2.60 GHz CPU, 8GB of memory, Windows 10 64-bit operating system. During the experiment, the composite access control model and the traditional single access control model share the same environment.

In this paper, CloudSim 4.0 is used to build cloud resource pool simulation platform. According to the object's access control requirements from the subject, a total of 1000 strategies are simulated. Each policy contains one to thousands of rules to verify the efficiency of the composite access control model. In the experimental test of access control effectiveness, I randomly select 1000 main access requests, including a number of malicious access requests and non-compliant access requests from subjects. Furthermore, the relationship between the average time of decision-making and the number of subject access requests are included, as shown in Fig. 3.

![Figure 3. Correctness Experiment of Access Control.](image)

The graph shows that the average time of decision-making increases as the number of requests increases. In this figure, the traditional ABAC model is compared with the CAC model. The CAC model will consume much more time than the ABAC does. However, their difference gradually stabilizes with the continuous increasing number of requests for subject’s access. The time consumed is only slightly more than the ABAC model. There is no much difference. This is mainly because the CAC model is essentially a combination of ABAC and RBAC. Thanks to the better optimization of CAC-RBAC stage, it does not consume too much time. As shown by the experimental results, the CAC model can indeed achieve the goal of access control without seriously affecting the decision-making time. The design of this model is correct.
Model Analysis

In the CAC model, the subject obtains the role after the access control judgment of subject attribute, object attribute and environment attribute. The permission limits the subject’s access ability based on assorted kinds of object attribute. And the permission supports for fine-grained access control. This double compound access control improves the security of the application.

Access control granularity is one of the cardinal factors that reflect the effect of access control. In the CAC model, both the subject and the object are represented by attributes. By defining the corresponding subject attributes and object attributes, the model can meet user's access control requirements for arbitrary granular resources and implement access control policies with different security requirements.

Conclusions

This paper designs a composite access control model based on attributes and roles. This model solves the problem that the RBAC model cannot satisfy the large-scale dynamic change of users. In the meantime, the model simplifies the complicated rights assignment and policy management in the ABAC model, especially in the aspects of strategy management and authority assignment. More importantly, the model meets the requirements of real-time dynamic cloud environment. After verification, it also meets the expected goal described in this paper.

There is still much room for improvement of the composite access control model designed in this paper. For example, when the model is assigning the role, can we simplify the process and optimize the time consumed? I have faith that in the near future, cloud computing will have a prosperous development and the model in this paper can be well enhanced and applied.

Acknowledgement

This research was financially supported by the National Natural Science Foundation (No: 61462069,61662056) and Inner Mongolia Natural Science Foundation (No: 2015MS0622,2016MS0609).

References


