Research on Usage Control Model with Delegation Characteristics Based on OM-AM Methodology

Zhiyong Zhang [1, 2], Lin Yang [3], Qingqi Pei[1], Jianfeng Ma[1]
[1] *Key Laboratory of Computer Network and Information Security, Ministry Of Education, Xidian University, Xi' an 710071, China*
[2] *Electron. Inf. Eng. Coll., Henan Univ. of Sci. & Technol., Luoyang 471003, China*
[3] *The Research Institute, China Electronic Equipment & Systems Engineering Corporation, Beijing 100039, China*
zhangzy@mail.haust.edu.cn

Abstract

$UCON_{ABC}$ is a basic framework of next generation access control policy Usage Control that is composed of Authorization-oBlige-Condition components, but so far it lacks of important delegation characteristic. The paper analyses the behaviors of delegation in UCON based on OM-AM engineering principles, presents a formalized usage control model with delegation features using BNF Extensions, called as $UCON_D$, and further articulates its hybrid architecture based on Client & Server Delegation Reference Monitors and relative key protocol functions. $UCON_D$ is an extension model of $UCON_{ABC}$ in the aspect of delegation authorization, and it resolves the delegation question of Usage Control Model. Moreover, we specify delegation procedure of an application for Digital Medium Resource Distribution System.

1. Introduction

The theory and technology of access control is a traditional research direction in the field of information and system security. From 1960s to 1990s, there are three representative access control policies as DAC, MAC and RBAC, as well as some corresponding models as matrix model, BLP and Biba model, Take-Grant, RBAC96, etc. Usage Control (abbr. UCON) was addressed based on some new application backgrounds and environments of information security at the beginning of this century, and it is a comprehensive framework combining access control, trust management and DRM to realize digital object privilege management efficiently [1, 2]. So far its research focus on $UCON_{ABC}$ construction and formal definition aiming at above mentioned different policies, but lacks of delegation property and related mechanisms [3, 4, 5]. One of contributions in this paper is constructing a fine-grained usage control delegation model with dynamic character and constraint rules, called as $UCON_D$. Besides, the other is applying OM-AM methodology, which is an engineering analysis approach with layer features, to specify the model objective, definitions, hybrid architecture, key protocol functions related to delegation realization mechanism based on an application for Digital Medium resource Distribution System.

2. Relative Research Works

This section presents some recent research related to UCON model and delegation mechanism, including $UCON_{ABC}$ framework and basic delegation properties, as well as OM-AM methodology dealing with access control policy implementation.

2.1. OM-AM Methodology and Engineering Principles

OM-AM methodology is an engineering principle for analyzing and designing large-scale information security system policy and model, proposed by Prof. Ravi. Sandhu firstly[6].The principle stands for Objective, Model, Architecture, Mechanism four layers in sequence, and differs from traditional top-down waterfall-style process of software engineering. Of above all, Objective layer presents the goals of security policies. Then, in Model layer, we need customarily articulate components of a model that should achieve

0-7695-2943-7/07 $25.00 © 2007 IEEE
DOI 10.1109/NPC.2007.103

above mentioned goals, including formalized definitions using predication logic or self-defined makeup language. With respect to Architecture layer, a framework, which is further divided into infrastructure and application sub-layers, is specified according to the model. For instance, there are familiar server-pull architecture, client-pull architecture, hybrid framework in RBAC application, and so on. The last, material realization mechanisms including key protocols and functions are represented in Mechanism layer, where feasible approach to meet the application requirements should be addressed in detail. The OM-AM methodology has been already analyzed for RBAC policy, Digital Rights Managements, etc.

2.2. UCON Framework and Delegation Objective

Traditional access control model and trust management resolve the questions of authorized user and anonymous user' s privilege assignments, delegation and access decision from respective different view. Their primary principles are based on Reference Monitor or other Trusted Computing Bases, entities' identifier, attributes and discrete context, realizing authorization and access decision, further controlling subject's access process and meeting the requirements of security objective.

Taking the demand of digital object security and DRM management into consideration, UCON policy was addressed combining authorization, obligation and condition. It is a policy-neutral control framework with continuity and changeability characters, and differs from conventional access control. The first, the model's changeability embodies the change of usage context including entities' attributes, temporal and dimensional condition. The second, these changes make it necessary that usage decision does happen at the whole usage procedure rather than only at the beginning of usage.

2.3. Delegation Properties

The basic idea of delegation is that active entity (user, process, agent, et. al.,) in application could grant some own permissions or roles to others, which can carry out some privileges and functions on behave of the former. For example, in the enterprise organization, somebody could delegate some permissions to other staffers and share privileges with them because of being absent or needing to collaborate with others. At the same time, he could also revoke these delegated-rights in need. The concepts related to delegation have delegator, delegated role or permission and delegatee.

Delegation has some important features as follows:

- Delegation Granularity: The unit of Delegation has three kinds as follow: Permission-based thin granularity [7], role-based medium granularity [8, 9], permission and role-based fat granularity proposed by ZHANG [10]. Thin granularity means that user could delegate the partial permissions of a role to delegtee, not just whole role. So granularity is depressed, and it meets the principle of least privilege, but brings about some non-integrity roles and leads to authorization complexity .For medium granularity, delegtor only could delegate role as a whole, thus delegtee would acquires entire permissions of role. Apparently, it sacrifices the principle of least privilege at some degrees. Fat granularity allows user to delegate own permission or role discretionarily, which is flexible compared with above two granularities besides of complex realization. With regard to delegation unit, it should be chosen according to applied system. For simplicity, the paper takes permission-based delegation granularity as example to articulate architecture and protocol functions.
- Delegation Step: It is subdivided into single-step delegation and multi-steps delegation. The former is that delegatee could not delegate role or permission to others further, and the latter allows delegatee to grant further, but in the condition revocation is more complicated.
- Delegation Temporal Limitation: Delegation is usually temporary, thus delegated permission and role having temporal periodicity characteristic. Beyond delegation time limitation, permission or role could be revoked.
- Delegation Revoking: The contrary operation of delegation is revoking which means that delegated roles or permissions are called off. Revoking mainly includes the following features, such as cascading revocation, non-cascading revocation, grant-independent revocation, grant-dependent revocation, system automatic revocation and user discretionary revocation.

3. $UCON_D$ Formalized Definitions

$UCON_D$ is an extension model framework having delegation capability based on $UCON_{ABC}$, and it remains two important intrinsic properties. It mainly includes delegation and usage decision, as Figure 1. The former is composed of delegator, delegtee, privilege and delegation, usage decision and test, shared resource constitutes the latter. For the goal of assuring accuracy of the model and further formalism validation, the paper represents formal definitions using BNF Extensions. On account of BNF Extension'

s flexible definition and being suitable to articulate framework, we do not adopt set theory and predication logic, or other self-defined makeup languages in the paper. The formalized definitions of $UCON_D$ are as follows.

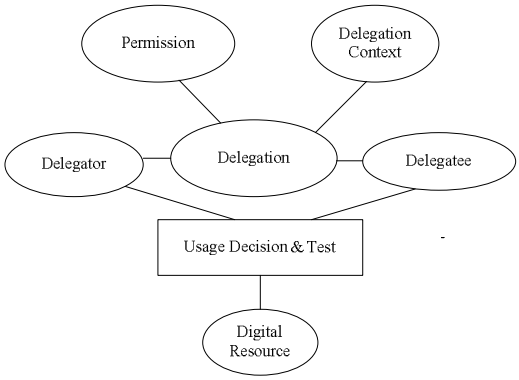

Figure 1. $UCON_D$ model with delegation characteristic

Definition1. ($UCON_D$ Model): It is a model of having delegation feature, including Delegation Entities, Attributes, Contexts, Rules, Verification, Decision, Test, Delegation, Revocation.

$UCON_D$ :: =<Entities><Contexts><Delegation><Dele-Verification><UsageDecision><UsageTest><Revocation>

Definition2. ($UCON_D$ Entities): Delegator, Delegatee, Shared Resource, Delegated Permission all belong to the entities of $UCON_D$. The relations among them are as Figure 1.

- Delegator: It is an actor for the goals of sharing resource, cooperative work, and temporary privilege transfer.
- Delegatee: It is an effector of acquiring long-term or temporary privileges to share resource with delegator.
- Resource: They are some shared digital data in open environment, including digital file, audio &video information, process, store, and web services, etc.
- Permission: Some delegated privileges include read, write, execute, copy, and abstract privilege.

<Entities>::=<Delegator>|<Delegtee>|<Resource>|<Per-mission>
<Delegator>::=<DlgtorID><DlgtorAttr><Permission>
<Delegatee>::=<DlgteeID><DlgteeAttr><Permission>
<Resource>::=<ResoID><ResoAttr>
<Permission>::=<Read>|<Write>|<Execute>|<Copy>|<Modify>|<Delete>|<Cascade_Delegation>|<Abstract_Perm>

Definition3. ($UCON_D$ Attribute): The attributes of $UCON_D$ are categorized into three kinds as follows:

- Delegator Attribute (abbr. Dlgtor_Attr): These attributes related to delegator are mainly identification (eg. UserID, role, group), security level, etc.
- Delegatee Attribute (abbr. Dlgtee_Attr): It is like as the above mentioned Dlgtor_Attr.
- Resource Attribute (Reso_Attr): Its scope is wider than the former two kinds, and here we denote some key attributes that affect usage decision, such as only-read attributes, non-sharing attributes, and usage cardinal number.

<Attributes>::=<DlgtorAttr>|<DlgteeAttr>|<ResoAttr>
<Dlgtor_Attr>::=<Status>|<Role>|<Security Level>
<Dlgtee_Attr>::=<Status>|<Role>|<Security Level>
<Reso_Attr>::=<Status>|<Security Level>

Definition4. ($UCON_D$ Rule): It denotes the primary principle that is satisfied in delegation decision.

- Permission Granularity Rule: It defines basic delegated unit including the kinds of whole permission and partial permission.
- Permission Collision Rule: It defines the non-concurrence rule of privilege.
- Delegation Step Rule: It decides whether delegated privilege can be further transferred or not.
- Delegation Time-Limit Rule: It is related to delegation temporal character (eg. delegation beginning and end time, delegation periods).
- Delegation Revocation Rule: It presents delegation drop modes, which are cascading revocation, non-cascading revocation, grant-independent revocation, grant-dependent revocation, system automatic revocation and user revocation.

Rules={Granularity, Collision, Step, TimeLimit, RevoMode}
Rule_Granu::=<Permission>|<Role>|<RolePermission>
Rule_Colli::=<Delegator>|<Delegatee>|<Rule_Granu>|<Mutex_Granu1>|<Mutex_Granu2>
Rule_Step::=<Single>|<Multi>
TimePeriod::= <Begintime>|<Endtime>
Rule_TimeLimit::={<TimePeriod>…}
Rule_RevoMode::=<Non_Cascade>|<Cascade>|<Delegtor>|<System>|<Granu_Independent>|<Granu_dependent>

Definition5. ($UCON_D$ Contexts): Contexts include current system environment, three kinds of delegation attributes and some conditions related to delegation rules, and are also used as evidences of delegation verification and test.

<Contexts>::=<Environment><Attributes><Rule-Related Conditions>

Definition6. (Permission Delegation): It is a procedure involved the delegator and delegatee, and written as a five tuple: (Dlgtor, Dlgtee, Reso, Permission, Context).Its semantic is that for the aim to share, collaboration , Dlgtor can delegate the

permission of Reso to Dlgtee in context environment, so Dlgtee acquires privilege and acts on behalf of delegator.

<Delegation>::=<Dlgtor><Dlgtee><Resource><Permission><Contexts>

Definition7. (Delegation Revocation): When the attributes of entities and context are in collision, delegated privilege will be dropped. Revocation mode could affect system's expenditure and efficiency. Revocation mode is similar to revocation rule in Definition 5.

<Revocation>::=<Dlgtor><Dlgtee><Reso><Perm><RevoMode><Step>

Definition8. (Delegation Verification): It decides whether the current delegation is permitted or not, and is realized by Delegation Reference Monitor that is the trust computing base of system. After Clint-Delegation Reference Monitor receives delegation request, it makes decision according to delegated certification and context, then sends the result (permission or forbiddance) to delegator.Here are two operations related to delegation rule: rule satisfaction operation is written as ▲, non-satisfaction operation written as ▼.

```
<DeleVerification>:: = Dlg_Veri
  <input>::=<Dlgtor><Permission><Contexts>
  <output>::=<Allow>|<Reject>
   {
     <C-DRM>: if Dlgtor_Attr && Dlgtee_Attr &&
    Reso_Attr && Permission ▲ rules then
              {send_result (Allowance);
               Delegation;}
            else
              send_result(Rejection)
            end;
    }
    End;
```

Definition9. (Usage Decision and Test): Usage control mainly embodies the two actions: usage decision and test. The former happens at the beginning of resource usage, the latter acts at the procedure of usage. They are also some parts of trust computing base that is executed by DRM.

- Usage decision: After receiving the delegation certification of C-DRM, Server-DRM makes decision to certification and notify the information (yes or not) to delegator.
- Usage test: On account of attributes being variability, S-DRM must test decision result periodically for attribute consistency.

```
<UsageDecision>:: = Usage_Deci
   <input>::=<Dlgtee><Resource><Dlgtee_Attr><Contexts>
   <output>::=<Allowance>|<Reject>
   {
    <S-DRM>: if Dlgtor_Attr && Dlgtee_Attr &&
  Reso_Attr && Permission ▲ Authorization&&
  Environment then
              {send_result (Allowance);
               Access to Resource;}
            else
              send_result(Reject)
            end;
   }
   End;

<UsageTest>::="Usage_Test"
  <input>::=<Dlgtee><Resource><Attributes><Time_Period>
  <output>::=<Allowance>|<Rejection>
    {
      for(i=0; ; TimePeriod)
      { <S-DRM>:<UsageDecision>; }
    }
   End;
```

4. Hybrid Architecture and Key Protocol Functions

The architecture of $UCON_D$ adopts Clint-Delegation Reference Monitor(C-DRM) and Server-Delegation Reference Monitor(S-DRM) hybrid pattern as Figure2. Client runs C-DRM that is Trusted Computing Base (abbr. TCB) of delegation verification, it also has a copy of general authorization assignments. C-DRM verifies delegation request according to related delegation contexts including constraint rules, then issues Delegation Certificate (abbr. DC) to delegatee and sends it to the delegation database of Server side, so fulfilling delegation process. Resource sever adopts S-DRM that is the TCB of delegation decision and test. According to entities' attributes, S-DRM test delegated permission validity periodically. If delegation could be invalid or expired, server pause the capability promptly. Delegatee could act on delegated capability through submitting to Server, thus access to resource. Besides, audit subsystem could record and track on the delegatee's usage of digital medium resource by DC.

Because resource storing patterns could be centralized, federal or distributed framework, S-DRM is also the same configuration. Here only gives common centralized resource architecture.

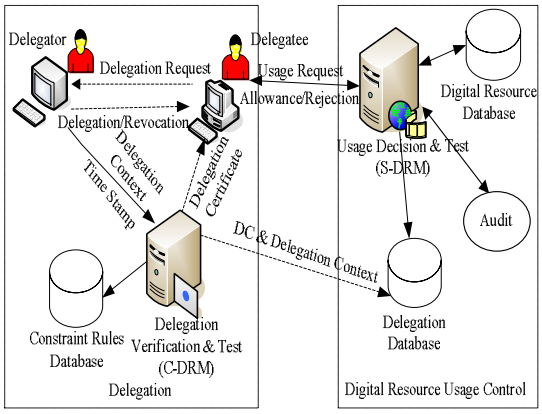

Figure 2. Hybrid architecture of usage delegation on centralized shared digital medium resource

The key protocol functions and related semantics of the model are illustrated in Table1, for example, delegation request function DeleRequest, delegation verification function DeleVerify; delegation answer function DeleAnswer; delegation function Delegation, revoke function Revocation, access decision function AccessDecision, delegation test function DeleValiTest. The foregoing five functions belong to C-DRM, others act on S-DRM.

Table 1. Key protocol functions

Protocol Functions	Related Semantics
DeleRequest(delegator, delegtee, resource, perm)	Delegatee requests for permission delegation from delegator.
DeleVerify(delegatee, delegator, perm, contexts)	Delegation verification is implemented on C-DRM integrating context.
DeleAnswer(delegator,delegatee,result)	C-DRM sends result to delegatee.
Delegation(delegator,delegatee, DC, timestamp)	Delegator and delegatee subscribe DC on C-DRM, then DC is also sent to Server side.
Revocation(delegator,delegatee, DC, timestamp)	Revoke DC of delegatee and notify S-DRM.
AccessDecision(delegatee, resource, DC, contexts)	S-DRM makes usage decision according to DC.
DeleValiTest(delegatee,resource, DC, attributes, timeperiod)	S-DRM test delegation validity in every period based on attributes of entities.

Considering heterogeneous platform and software migration of $UCON_D$ application, the above mentioned key functions were realized in Java, and defined main attributes and methods of classes, such as Delegator, Delegatee, Resource and Context. Here represents delegation process between users: User_Dlgtor acquires some permission in advance, User_Dlgtee requests permission(s) for sharing and accessing to digital medium resource. A material process is as follows:

1. User_Dlgtee calls function DeleRequest (User_dlgtor, User_dlgtee, resource, perm) on C-DRM, and send the request of permission delegation of resource to User_Dlgtor.
2. User_Dlgtor calls DeleVerify (User_dlatee, User_dlgtor, perm, context) on C-DRM to verify the request in terms of delegation context.
3. User_Dlgtor calls deleAnswer (User_dlgtee, User_dlgtor, result) on C-DRM to send verification result to User_Dlgtee, then calls Delegation(User_dlgtor, User_dlgtee, DC, timestamp) subscribes Delegation Certification with User_Dlgtee, at the same time the DC is sent to delegation database of Server side.
4. When User_Dlgtee begin to access to shared digital medium resource, AccessDecision (User_dlgtee, resource, DC, contexts) acts on S-DRM for usage decision; in the whole procedure of usage, DeleValiTest (User_dlgtee, resource, DC, attributes, timeperiod) test delegated permission validity in every given periods by system and delegator.
5. If the attributes of User_Dlgtee, User_Dlgtor, or resource change, system or User_Dlgtor could call the function Revocation (User_dlgtee, User_dlgtor, DC, timestamp) to revoke delegation in line with revocation modes and rules, then notifies Server side.

Here delegator also could delegate its capability discretionarily for sharing resource, not through delegatee's request, thus delegation step 1 could be omitted.

5. Conclusions

$UCON_D$ is a fine-grained usage control delegation reference model. This model not only maintains $UCON_{ABC}$ continuity and changeability characters, but also supplies its lack of delegation mechanism, so UCON framework is more mature. According to OM-AM analysis methodology, the paper proposes delegation objective, a new model, hybrid architecture, and material mechanism including key protocol functions, as well as presenting practical application of $UCON_D$. Our future researches focus on visual modeling integrating software engineering [11], as well as UCON management model's formal definitions and specification.

6. Acknowledgement

I am grateful to three anonymous reviewers for their helpful suggestions and comments. The paper is supported by General Program of National Nature Science Foundation of China (Grant No. 60672112), General Program of National Nature Science Foundation of China (Grant No. 60573036), Henan Univ. of Sci. & Technol. Research Foundation for Young Scholars (Grant No.2005QN019).

7. References

[1].Jaehong Park, and Ravi Sandhu, "Towards Usage Control Models: Beyond Traditional Access Control", *The Proceedings of 7th ACM Symposium on Access Control Models and Technologies*, 2002, June.

[2].Ravi Sandhu, and Jaehong Park, "Usage Control: A Vision for Next Generation Access Control", *The Proceedings of Mathematical Methods, Models, and Architectures for Network Security Systems2003*, St. Petersburg, Russia, 2003.

[3].Ravi Sandhu, and Jaehong Park, "The $UCON_{ABC}$ Usage Control Model", *Transaction on Information and System Security*, Vol.7, No.1, pp.128-174.

[4].Zhao Baoxian, and Qin Xiaolin, "A Survey on the Database Access Controls", *Computer Science,* 2005, Vo1. 32, No.1, pp. 88-91. (in Chinese)

[5].Yuan Lei, "Research On Usage Control Model", *Computer Engineering,* 2005, Vol.31, No.12, pp.146-148. (in Chinese)

[6].Ravi Sandhu. "Engineering Authority and Trust in Cyberspace", *The OM-AM and RBAC Way*, *The Proceedings of ACM Workshop on Role Based Access Control 2000*, Berlin, Germany.

[7].Xinwen Zhang, Sejong Oh, and Ravi Sandhu, "PBDM: A Flexible Delegation Model in RBAC", *The Proceedings Of SACMAT'03*, omo, Italy, 2004, June 2-3.

[8].Ezedin Barka, and Ravi Sandhu, "A Role-Based Delegation Model and Some Extensions", *The Proceedings of 16th Annual Computer*, Sheraton New Orleans,2000.

[9].SangYeob Na, and SuhHyun Cheon, "Role Delegation in Role-Based Access Control", *The Proceedings of ACM RBAC2000*, 2000.

[10].Zhang Zhiyong, and Pu Jiexin, "Permission-Role Based Delegation Model and Object- Oriented Modeling", *The Proceedings of China National Open Distributed and Parallel Computing Symposium 2004*, Beijing, China, 2004, Nov.18-20.

[11].Zhang Zhiyong, and Pu Jiexin, "Delegation Model for CSCW Based on RBAC Policy and Visual Modeling", *The Proceedings of the 11th Joint International Computer Conference 2005*, Chongqing, China, 2005, Nov.10-12.