UCOND：Usage Control委托模型及关键技术研究

张志勇　普杰信　黄涛
河南科技大学电子信息工程学院　河南　471003

摘要：本文基于委托基本特性的研究，分析了它在UCON体系框架中的具体表现，提出了一种具有委托特征的UCON模型—UCOND，同时给出了客户端-服务器端访问监控器相结合的体系架构及关键的核心函数。UCOND补充了UCONABC基本框架，解决了使用控制中的委托授权问题，使其更具有完备性和可操作性。

关键词：　使用控制；UCONABC；委托；UCOND

0　引言

Usage Control（UCON）是近两年随着信息与网络安全不同应用环境的需要在访问控制领域中所提出的一种全新的，结合访问控制、信任管理和数字版权管理等不同应用背景，实现数字化对象特权管理较为完备的体系架构及其相关理论模型。迄今UCON研究主要集中在基本模型UCONABC的构建和在不同应用背景下的形式化描述，而对于其中特权委托基本特征及其关键技术等并未涉及。本文对委托的基本特性和UCON中特权委托等关键技术进行了深入研究，提出了具有委托特征的UCOND模型，包括可实现的C-DRM和S-DRM相结合的体系架构以及关键核心函数等。

1　UCON策略与模型

UCON模型结合目前数字化对象安全和DRM管理的需求，结合授权、职责和条件所提出的一种具有访问使用连续性和主客体对象可变性的全新使用控制体系架构。其中连续性主要体现在主体的访问具有一定的时间持续性和实时性，从而访问决策不再仅出现在访问之间，它将伴随主体的整个使用过程，这不同传统的访问控制；可变性主要表现在主客体的相关属性随着访问过程中的不同因素，如时间和空间等会发生相应的变化，这一点也不同于传统访问控制中属性的变化主要是由于集中式管理行为的作用，如更新和撤销属性等。

2　委托授权基本特性

（1）委托授权粒度：委托的基本单位（粒度）有以下几种：细粒度是指允许用户将角色中的部分许可委托给另一用户，而不只是角色的整体委托；中粒度是指用户只能将自身的角色整体委托给其他用户，从而使其获得该角色所具有的全部许可权限，在某种程度上违背了最小特权原则；粗粒度委托是用户可以任意将自身的角色和（或）许可委托给他人，它比前两种粒度灵活，然而实现时较为复杂。关于委托粒度需要根据实际应用系统的需求折衷选择合适的委托授权基本单位。

（2）单步委托或多步委托：单步委托是指受托者不可以进一步地将委托角色或许可再次委托给其他用户；多步委托则允许受托者进一步实施委托，但撤销委托将变得复杂和困难。

（3）委托撤销：委托的逆操作称为撤销（Revoke），它完成被委托角色或许可的回收。撤销的主要方式有级联撤销、非级联撤销，独立于授权的撤销、非独立于授权的撤销等。

3　UCOND模型及形式化描述

UCOND模型是一种具有普遍特征的特权委托框架，是UCONABC在委托技术上的扩展模型，如图1所示。它保持了UCONABC的可变性和连续性两个主要特征，主要由委托授权和使用控制两部分构成。其中前者包括委托者、受托者、特权和委托上下文等组件；后者由使用决策与检测、共享信息资源组成。为了保持理论模型的精确性、无二义性，本文采用扩展的巴克斯范式对UCOND进行了形式化描述。

本文得到教育部科学技术重点资助项目（No:03081）和河南科技大学青年基金项目（No:2005QN19）的资助。
作者简介：张志勇（1975-），男，硕士，讲师。研究方向：RBAC与访问控制，智能决策支持系统；普杰信（1959-），男，博士，教授，院长。

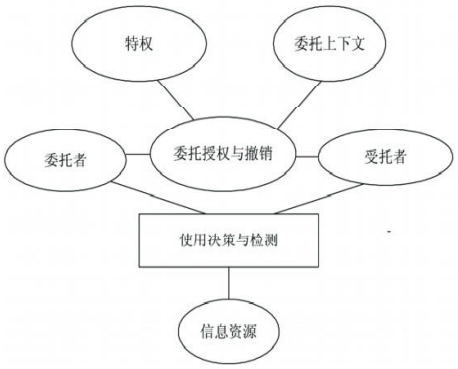

图1 UCOND模型

定义1（UCOND 实体）： UCOND 模型中的相关实体是：委托者、受托者、共享资源和委托特权。

（1）委托者：为达到资源共享、协同工作和权利暂时转移而进行特权转让的动作实施者。

（2）受托者：享受资源共享而长期或暂时获得特权的动作受动者。

（3）共享资源：开放式环境下可以共享的信息，包括文件、进程、存储等等。

（4）委托特权：针对共享资源可以转移的访问权利，如读、写、执行、拷贝等等，以及抽象的权利等。

<Delegator>::=<DlgtorID><DlgtorAttr><Permission>

<Delegatee>::=<DlgteeID><DlgteeAttr><Permission>

<Resource>::=<ResoID><ResoAttr>

<Permission>::=<Read>|<Write>|<Execute>|<Copy>|<Modify>|<Delete>|<Cascade_Delegation>|<Abstract Perm>

定义2（UCOND 属性）： UCOND 模型中的属性主要表现在委托者属性、受托者属性和资源属性三个方面。这些属性具有访问过程可变性，它们的变化必将引起访问决策的更新。

(1)委托者属性(Dlgtor_Attr):和委托者相关的属性主要有委托者的身份特征(如角色)、安全密级等等。

(2)受托者属性(Dlgtee_Attr)：和委托者的属性相似。

(3)资源属性(Reso_Attr)：资源的属性比较广泛，这里主要指影响访问决策的关键属性，如资源的只读性、非共享性、访问基数限制性。

<Dlgtor_Attr>::=<Status>|<Role>|<Security Level>

<Dlgtee_Attr>::=<Status>|<Role>|<Security Level>

<Reso_Attr>::=<Status>|<Security Level>

定义3（委托上下文）： UCOND 委托上下文包括当前系统环境、委托属性和委托规则。委托上下文可以作为委托审计的依据。

<Context>::=<Environment><Attributes><Rules>

定义4（委托规则）：委托决策时必须满足的若干条件。

（1）委托特权粒度规则：定义被委托特权的基本单位，主要分为整体特权和部分特权两种。

（2）委托特权冲突规则：定义委托特权之间不可同时存在的规则。

（3）委托撤销规则：定义委托撤销的特征，如级联撤销、非级联撤销，独立于授权的撤销、非独立于授权的撤销，系统自动撤销和用户撤销等。

Rule_Granu::=<Permission>|<Role>|<Role-Permission>

Rule_Colli ::=<Delegator>|<Delegatee>|<Rule_Granu>|<Mutex_Granu1>|<Mutex_Granu2>

Rule_Step::=<Single>|<Multi>

Rule_RevoMode ::=<Non_Cascade>|<Cascade>|<Delegtor>|<System>|<Granu_Independent>|<Granu_dependent>

定义5(特权委托)：委托过程是委托者和受托者实体间的一次授权过程，可以用五元组表示为(Dlgtor, Dlgtee, Reso, Permission, Context)。其语义为在共享的Context环境下Dlgfor将Reso的Permission特权委托转移给Dlgtee，Dlgtee从而享有Dlgtor拥有的权利，和他进行协同工作或代表他实施权利。

<Delegation> ::=<Dlgtor><Dlgtee><Reso><Perm><Contexts>

定义6(委托撤销)：当实体的属性和委托上下文发生冲突时而采取的特权回收。撤销方式和委托步将影响撤销时的系统开销和效率。

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

定义7(委托判定、使用决策与检测)：委托判定是委托者决定是否转让授权给其他用户。而使用控制又包含两种动作：决策是在受托者初始使用资源时实施；检测在资源使用中进行，从而满足UCONABC的连续性和实时性。判定、决策与检测由客户端和服务器端的访问监控器RM完成，它是整个系统的可信计算基。

（1）委托判定：C-DRM在收到委托消息后，根据委托证书的相关信息、上下文环境而进行决策，最后将决策结果(允许/禁止)发送给委托者。

（2）使用决策：S-DRM在收到C-DRM委托证书后，对委托证书进行决策，决策结果(承认/否认)发给委托者。

（3）使用检测：在委托特权行使过程中，由于UCOND属性的可变性，S-DRM需要周期性地进行决策结果检测，使其和当前属性相一致。

4 UCOND体系架构与关键核心函数

在UCOND 体系架构中客户端采用C-DRM，它是客户端委托判定的可信计算基。C-DRM根据委托规则判定委托，完成特权委托，签发委托协议证书Delegation Certificate(DC)发往服务器端。服务器端根据DC和委托上下文等进行DC有效性的认证，并告知delegator。委托证书作为日后审计资料存储

在审计系统中。信息资源服务器端采用S-DRM，它是服务器端使用判定与检测的可信计算基。S-DRM根据相关属性，周期性地检测委托的有效性。委托失效时，立即中止特权的使用。这里给出通常所采用的集中式服务器体系架构，如图2所示。

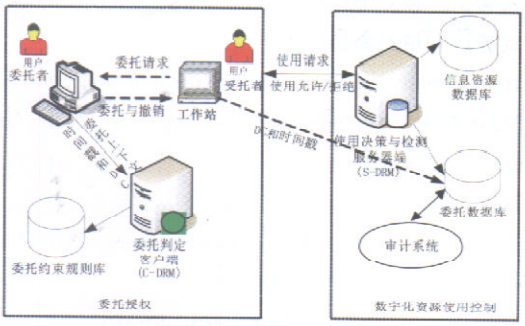

图2 UCOND委托体系架构

UCOND 模型的关键函数和语义如表1所示：委托请求函数DeleRequest；委托判定函数DeleVerify；委托应答DeleAnswer;委托授权Delegation;委托撤销Revoke;其中前三个函数属于C-DRM端函数，后两个属于C-DRM和S-DRM函数。使用决策函数UsageDecision;检测函数UsageTest，这两个函数属于S-DRM函数。

5 结论

本文提出的UCOND是一个具有委托特性的使用控制模型，它不仅保持了UCONABC的连续性和可变性特征，并弥补了UCONABC研究中缺少委托授权策略和机制的不足，使得UCON框架更加完备。进一步地丰富该模型的语义，构建UCONABC管理模型将是后续的研究工作。

参考文献

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

[2]Ravi Sandhu, Jaehong Park. The UCONABC Usage Control Model. Transaction on Information and System Security.Vol.7. No.1.Feb.2004.

[3]赵宝献,秦小麟.数据库访问控制研究综述.计算机科学,2005.

[4]袁磊.使用控制模型的研究.计算机工程.2005.

[5]Zhang Zhiyong, Pu Jiexin. Permission-Role Based Delegation Model and Object-Oriented Modeling.Proc. of National Open Distributed and Parallel Computing Symposium2004. Nov.18-20. Beijing, China.

表1 UCOND 委托关键函数(协议)

主要参与对象	关键核心函数（协议）	语义
Dlgtee, C-DRM	DeleRequest(delegator,delegtee,resource,permission)	delegatee向delegator发送resource permission的委托请求。
Dlgtor, C-DRM	DeleVerify(delegatee,delegator,permission,context)	结合context环境C-DRM进行委托判定。
Dlgtor, C-DRM	DeleAnswer(delegator,delegatee,result)	判定结果发给delegatee。
Dlgtor, Dlgtee, C-DRM ,S-DRM	Delegation(delegator,delegatee,DC,timestamp)	delegator和delegatee在C-DRM签署委托证书DC，并发给S-DRM.
Dlgtor, Dlgtee, C-DRM ,S-DRM	Revoke(delegator,delegatee,DC,timestamp)	收回delegatee享有的委托证书DC，并告知S-DRM.
S-DRM	UsageDecision(delegatee, resource, DC,context)	S-DRM针对委托证书进行使用决策.
S-DRM	UsageTest(delegatee,resource,DC,attributes,timeperiod)	S-DRM根据attributes进行周期性决策检测.

UCOND:Research on Usage Control Delegation Model and Key Technology

Zhang Zhiyong,Pu Jiexin,Huang Tao
Electron.Inf.Eng.Coll.,Henan Univ.of Sci.&Technol,Henan,471003

Abstract:The paper analyses the behavior of delegation in UCON,presents a usage control model with delegation UCOND,and describes its framework with Client-Server Reference Monitors and key core functions.UCOND supplements UCONABC,and resolves the delegation of Usage Control having more mature and applied features.

Keywords:Usage Control;UCONABC;Delegation;UCOND