基于角色的层次受限委托模型

刘正涛[1,3]，毛宇光[1,2]，王建东[1]，叶传标[3]

(1. 南京航空航天大学信息科学与技术学院 南京 210016; 2. 南京大学计算机软件新技术国家重点实验室 南京 210093;
3. 三江学院计算机科学与工程系 南京 210012)

【摘要】角色委托是RBAC模型需要支持的一种重要安全策略。基于构件化的思想，在基于角色访问控制模型基础上，提出了一个受限的层次角色委托模型，该模型分别在时间约束、部分委托约束、角色依赖约束、角色冲突等方面对委托进行了限制。给出了委托授权时的冲突检测算法与用户所拥有权限的计算算法及该模型的一个应用实例。

关 键 词 访问控制; 构件; 委托; 层次
中图分类号 TP311 **文献标识码** A doi:10.3969/j.issn.1001-0548.2010.01.026

Role-Based Constraint Hierarchy Delegation Model

LIU Zheng-tao[1,3], MAO Yu-guang[1,2], WANG Jian-dong[1], and YE Chuan-biao[3]

(1. College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics Nanjing 210016;
2. State Key Laboratory for Novel Software Technology, Nanjing University Nanjing 210093;
3. Department of Computer Science and Engineering, Sanjiang University Nanjing 210012)

Abstract Delegation is an important security policy supported by role based access control (RBAC) model. Based on the idea of components and role based aceess control model, this paper presents a constraint hierarchy Role-Based delegation model with time constraint, partial delegation constraint, roles dependency constraint, and roles conflicts constraint. The paper also explores some issues including conflicts examining algorithm and rights computing algorithm. In the end, an application example is provided using this model.

Key words access control; components; delegation; hierarchy relation

RBAC的主要思想是将授权和角色联系在一起，可以分配角色到用户，使用户通过角色间接地访问资源。目前，RBAC96[1]是得到信息安全领域广泛接受的RBAC参考模型。为方便权限管理，RBAC96模型引入了角色层次，角色的继承允许某一些角色定义为另一角色的子角色，通过角色之间的继承关系，间接地拥有其子角色所定义的权限。这样可以简化系统管理员的操作，减少了出错的可能性。当前的一些研究工作是在RBAC的基础上扩展其表达能力，使其更符合实际的应用状况。其中基于角色的委托受到了广泛的关注。

委托是一种重要的安全策略，其主要思想是系统中主动实体将全部权限或部分权限委托给其他主动实体，以便后者以前者的名义执行相关的工作。如在某用户出差或生病等情况下，为了能继续执行他负责的工作，需要将其拥有的权限委托给其他人，用户回来后，需将被委托权限的用户访问权限撤消。

限制从一开始就是RBAC研究的主要内容之一，研究者通过扩展RBAC的限制来增强它的表达能力，以适应不同情况下的权限管理。早期对限制的研究主要集中在权责分离上，稍后的研究内容扩展到其他方面，包括角色的用户数目限制、时间和依赖性限制、角色的使用限制以及限制的形式化描述语言等[2-6]。

在基于角色的委托模型方面，文献[7]提出了一个基于角色的委托模型，该模型支持用户到用户的角色委托，并且非形式化地提出了一些扩展，包括取消委托、部分委托、多步委托等。文献[8]提出了PBDM(permission-based delegation model)角色委托模型，它最大特点的是支持部分委托和角色到角色的委托。其中，部分委托是通过用户创建委托角色，并将其拥有的角色中的权限赋予新的角色实现。实现方法上有一定的创新，但是该方式导致模型的使用和管理变得更加复杂。文献[9]描述了一个基于规

收稿日期：2008 – 07 – 11； 修回日期：2009 – 04 – 27
基金项目：国家863重大项目(2006AA12A106)
作者简介：刘正涛(1975 –)，男，博士生，主要从事信息安全、数据库系统及理论方面的研究.

则的框架处理角色的委托和撤消，通过引入委托关系支持角色分层和多步委托。文献[10]研究了用户到用户的权限委托时的限制问题，其中包括临时性限制、时序依赖性限制等，但是该模型是基于扁平角色的，不支持角色继承。文献[11]给出了PDACDM(periodicity discretionary access control delegation model)模型，描述了如何在基于周期时间限制的自主访问控制模型上实施细粒度的权限委托。文献[12]综合RBAC和信任管理各自的优势，提出了一个适合开放式环境的委托授权模型，较好地解决了上述权限传播控制问题，有助于精确实现Internet环境下网络资源的访问控制，为构建面向互联网资源共享的虚拟计算环境提供了有力的授权技术支持。但该模型因过于复杂，不适合一般的应用系统。

总的来说，RBAC模型的研究工作已经比较成熟，而其中限制的研究工作得到了充分的重视，基于角色的委托模型限制方面已经有了初步的研究。本文在前人研究的基础上，从实际应用出发，提出了一个层次的基本委托模型，在该基本模型的基础上，分别在时间约束、部分委托、依赖委托、角色冲突约束方面进行相应的扩展，给出了一个完整的受限层次委托模型。最后给出了授权是否冲突的检测算法及一个用户所能拥有的权限集合的算法，为实际应用该模型奠定了基础。

1　基本委托模型

RBDM(role-based delegation model)是一个简单的委托模型，该模型基于RBAC96模型。在该模型中，角色之间有继承关系。

1.1　RBAC96模型

定义 1　RBAC96模型主要包括以下6个部分：

(1) U、R、P分别表示用户集、角色集、权限集。

(2) PA$\subseteq P\times R$表示一个多对多的权限到角色的分配关系。

(3) UA$\subseteq U\times R$表示一个多对多的用户到角色的分配关系。

(4) Users: $R\rightarrow 2^U$表示角色集到用户集的映射函数，Users(r)={U|(U, r)∈UA}。

(5) Permissions：$R\rightarrow 2^P$表示角色集到权限集的映射函数，Permissions(r)={P|(P, r)∈PA}。

(6) RH$\subseteq R\times R$表示角色集R上的偏序关系，称为角色层次关系，(r_1,r_2)∈RH表示角色r_1继承角色r_2的所有权限，记做$r_1\leqslant r_2$。

1.2　基本层次委托模型

定义 2　基本委托模型包括以下几个部分：

(1) Users_O(r)、Users_D(r)表示委托成员与被委托成员。其中委托成员为将自己的角色权限委托给其他用户的用户，被委托成员为接受权限的用户。

(2) UAO表示一个多对多的委托成员与角色的分配关系，UAD表示一个多对多的被委托成员与角色的分配关系。

(3) UA = UAO∪UAD，UAO∩UAD=ϕ 。

(4) Users_O(r)={U|(U,r)∈UAO}，Users_D(r)={U|(U,r)∈UAD}。

(5) 公有委托权限集Pub_PD，Pub_PD$\subseteq P$，且其权限可以被委托。

(6) 公有委托权限p^+，任意p^+∈Pub_PD。

(7) 私有委托权限集，Pri_PD，Pri_PD$\subseteq P$，且其权限不可被委托。

(8) 私有委托权限p^-，p^-∈Pri_PD。

(9) Pub_PD∩Pri _PD=ϕ，Pub_PD∪Pri_PD=P。

(10) 公有委托PD：PD$\subseteq U\times U$，即(u_1,u_2)∈PD表示用户u_1将权限委托给u_2，记作$u_1<u_2$。若$u_1<u_2$，则 $\forall p^+\in$Pub_PD(u_1)$\Rightarrow p^+\in$Pub_PD(u_2)，且 $\forall p^-\in$Pri_PD(u_1)$\Rightarrow p^-\notin$Pri_PD(u_2)。

委托模型具有以下特点：

(1) 同角色的委托用户之间不允许委托，因为这样是无意义的；同角色的委托用户拥有相同的权限，他们之间的权限是相同的，因此，不需要委托获取相关权限。

(2) 该模型在实现用户与用户之间的委托时，委托用户会将本身具有的所有公有委托权限委托给被委托用户。

(3) 在撤回委托关系之前，被委托用户将一直拥有所委托的权限，直到该委托关系被终止。

(4) 在获得委托权限后，被委托用户不能将所委托的权限再次委托给其他用户。

(5) 不能继承委托权限。因为在一般情况下，继承是高级用户继承低级用户的权限，而委托不同，委托往往是高级用户将权限委托给低级用户或将权限委托给同级用户。

2　受限层次委托模型

2.1　相关约束的定义

定义 3　时间约束：

(1) 委托时间T是一个时间段，可以是周期时间，也可以是持续时间。T_o表示RABC96中的时间函

数，T_d为委托时间$T=T_o\cap T_d$。如委托用户的时间周期为工作日的8:00~17:00，其委托时间到2008-5-1日为止，则被委托用户的使用权限为到2008-5-1日为止的工作日的8:00~17:00。

(2) 委托角色UAD→T是每一个委托到一个时间周期的函数映射。

时间约束最大的优点是在时间截止时，系统能自动回收被委托用户的委托权限，减少委托者的工作，在时间上保证了委托权限的安全性。

定义 4 角色依赖约束：

(1) 激活依赖是一个集合，其中的任意元素为ua或～ua，ua∈UAO，分别表示依赖的处于激活状态的用户、角色对或非激活状态的用户、角色对，激活依赖表示为DEP。

(2) active(dep)={ua|ua∈dep}为得到依赖的激活状态的用户、角色对集合。

(3) inactive(dep)={ua|~ua∈dep}为得到依赖的非激活状态的用户、角色对集合。

(4) ∀ dep:DEP•active(dep) ∩inactive(dep)= ϕ ∧dep =active(dep)⋃{～x∈inactive(dep)}。

定义中ua=(u_1, r_1)形式的元素表示，使用该委托角色，需要保证在发生请求时角色r_1处于被用户u_1激活的状态，而ua形式的元素正好相反。定义中，约束(4)表示任意激活依赖dep仅由定义中两种形式的用户、角色对组成，并且它们不相交，即不存在既依赖某个用户激活特定角色，同时又依赖于某个用户非激活特定角色这种自相矛盾的情况。

定义 5 角色互斥约束：

(1) 静态角色互斥约束SSD为一个二元组(r_1,r_2)，静态角色互斥约束指满足静态职责分离约束的两个角色不能同时指派给同一个用户，即($\forall u\in U$) $\forall(r_1,r_2)\in R(u)\rightarrow r_1,r_2\notin$SSD。

(2) 动态角色互斥约束DSD为一个二元组(r_1,r_2)，动态角色互斥约束指互斥的角色可以分配给同一个用户，但该用户在一个会话中不能同时激活这些互斥的角色。即($\forall$u $\forall(r_1,r_2)\in S(u)\rightarrow r_1,r_2$ ∉DSD。

定义 6 多步委托约束：

(1) 委托深度约束，最大委托深度D为允许继续向下委托的最大步数。对于不能被委托的角色，其最大委托深度D=0；反之，能被向下代理的角色的最大委托深度D≥1。当委托用户的角色实现一次委托时，D的值减少1，当D的值为0时，该角色不允许继续进行委托。

(2) 委托宽度约束，最大委托宽度W指委托用户允许将权限委托给其他不同用户的总数量。对于不能被代理给其他的角色，其最大委托宽度W=0；反之，能被代理给多个用户，其最大委托宽度W≥1。

(3) 本次可委托角色集RD⊆R，即对于任意一个委托(u_1,u_2)，可以委托的角色权限为RD⊆ R(u_1)。

2.2 层次受限委托模型

定义 7 受限委托模型：

(1) 受限委托表示为一个5元组(u_1,u_2,t,rd,dep)，其中u_1，$u_2\in U$，$t\in T$，rd∈R_D，dep∈DEP，表示为DT，用dtt、dtdep等分别表示dt∈DT的5个分量。

(2) 两个依赖委托dt_1、dt_2相等当且仅当它们的5个分量分别相等，表示为dt_1=dt_2。

(3) 委托集就是由委托组成的集合，表示为DTS。

(4) ∀dt_s:DTS，∀dt_1, dt_2: DT, dt_1, $dt_2\in dt_s\leftrightarrow dt_1\neq dt_2$，表示依赖委托集中针对同一个用户、角色对至多有一个委托。

(5) uad∈UAD，dt_s∈DTS，map(dt_s,uad)={dt|dt_{uad}=uad}，函数返回某依赖委托集中做作用于特定被委托用户和委托角色对的依赖委托集合。

(6) ∀dt_s，如果委托角色与该用户的角色冲突，则$dt_s\notin$DTS。

依赖委托中的dep表示该角色委托需要满足的常规角色激活依赖。函数map返回一个依赖委托的集合，集合中的元素是dt_s中作用于uad的委托。根据前面的性质可知，该集合或者为空或者只有一个元素。

使用委托深度与宽度解决了在角色委托的过程中无限地进行委托导致的权限使用失控的问题，使用可委托角色集RD实现了层次委托中的部分角色委托问题。

3 授权算法

算法 1 委托授权冲突检查算法。

INPUT：u_1，$u_2\in U$

OUTPUT：Yes授权成功，No授权失败

BEGIN

IF $dt_{rd}\cap R(u_2)\neq\phi$

RETURN No; //无需进行委托

ELSE IF ($dt_{rd}\cup R(u_2)$)$\cap SSD\neq\phi$

RETURN No; //违反角色静态约束

ELSE IF Depth≥D–1 //Depth当前授权路径的

长度

RETURN No;　　//超越最大深度

ELSE IF Width≥W–1　//Width为当前授权的宽度

RETURN No;

ELSE

RETURN Yes;

END

委托检测冲突算法主要应用于进行委托授权时，主要作用是检测该次委托是否有冲突。

算法 2　计算一个用户所拥有的角色集合。

INPUT：$u_1 \in U$

OUTPUT：P

BEGIN

P=ϕ;

P=$P \cup P$(UA(u_1)); //本身拥有的权限集合

$P = P \cup P$(RH(u_1));　//继承得到的权限

$P = P \cup P$(DTS(u_1))+;　//通过委托得到的公有权限;

RETURN P;

END

算法2说明一个用户本身拥有的权限集合通过以下过程得到：首先通过UA获取该用户的所有角色集，然后再通过每个角色的权限集获取每个角色的权限集合；通过委托得到的公有权限集合过程为先通过集合DTS获得用户u_1所有的委托角色，再获取这些角色的公有委托权限集合。算法2返回的权限集合是一个具有时间约束、依赖约束的权限集合。

当一个用户的会话所拥有的权限集合时，必须考虑用户通过委托得到的权限集合的限制条件是否违反了角色之间的动态约束，如果违反将不能行使该角色的相应权限；如果该用户的某个角色与其他角色有依赖关系，则需要看其他角色的激活状态，只有依赖角色的为激活状态，该角色的权限才能被激活；如果通过委托得到的角色不在委托时间范围内，该角色的相应权限也不能得到激活。

4　模型应用

学员关系管理系统是为某航空公司的培训学校开发的一套以客户关系管理思想为基础的系统，该系统包含市场管理、招生管理、学员管理、学员关怀等模块。系统在权限管理时实现了委托基本模型及时间与层次的约束。实现的委托界面如图1所示。

图1　委托实现

在该系统中，委托模型中的撤消策略分为3种：

(1) 系统管理员撤消：无论何种原因，系统管理员都有权执行撤消的操作。

(2) 基于时间限制的委托撤消：指当授权时间超过有效时限时，系统自动地撤消。这种方式是一个自动触发的过程，有效地减轻了管理人员的负担。

(3) 委托用户主动地撤消：委托用户可以根据需要对自己委托的权限进行撤消。权限的撤消可以全部撤消，也可以根据需要对部分角色的委托权限进行撤消。

5　结　论

委托是RABC模型需要支持的一种重要的安全策略，本文提出了一个较为完整的基于角色受限层次委托模型，分别在时间约束、部分委托、依赖约束、角色冲突约束方面进行了相应的扩展；给出了一个在委托授权时检测冲突的算法与一个用户所拥有的权限集合计算算法；最后给出了该模型的一个应用实例，并讨论了该委托模型的委托撤消问题。与参考文献的相关委托模型相比，本文中提出的模型符合构件化的思想，在用户使用该模型时，可以先建立基本模型，然后根据需要对模型在多方面进行扩展；同时，该模型简单，容易使用，用户可以根据实际的需要，运用该模型来实现系统的权限管理。

参　考　文　献

[1] SANDHU R. Rationale for the RBAC96 family of access control models[C]//ACM Workshop on Role-Based Access Control. New York: ACM Press, 1996: 38-47.

[2] BERTINO E, BONATTI P A, FERRARI E. TRBAC: a temporal role-based access control model[J]. ACM Trans on Information and System Security, 2001, 4(3): 191-233.

[3] JOSHI J B D, BERTINO E, GHAFOOR A. Temporal hierarchy and inheritance semantics for GTRBAC[C]//Proc of the 7th ACM Symp on Access Control Models and Technologies. New York: ACM Press, 2002.

[4] JOSHI J B D, SHAFIQ B, GHAFOOR A, et al. Dependencies and separation of duty constraints in GTRBAC[C]//Proc of the 8th ACM Symp on Access Control Models and Technologies. New York: ACM Press, 2003: 51-64.

[5] 田敬东, 何再朗, 王向东, 等. 基于角色的强制访问控制模型研究[J]. 电子科技大学学报, 2006, 35(6): 950-952.
TIAN Jing-dong, HE Zai-lang, WANG Xiang-dong, et al. Research on access control of role-based MAC[J]. Journal of University of Electronic Science and Technology of China, 2006, 35(6): 950-952.

[6] TAN Liang, ZHOU Ming-Tian. Implementing discretionary access control with time character in Linux and performance analysis[J]. Journal of Electronic Science and Technology of China, 2006, 3(1): 274-280.

[7] BARKA E, SANDHU R. A role-based delegation model and some extensions[C]//Proc of the 23rd National Information Systems Security Conference. Baltimore: NIST, 2000: 101-114.

[8] ZHANG X W, OH S, SANDHU R S. PBDM: a flexible delegation model in RBAC[C]//Proc of the 8th ACM Symp on Access Control Models and Technologies. New York: ACM Press, 2003: 149-157.

[9] ZHANG L, AHN G J, CHU B T. A rule-based framework for role-based delegation and revocation[J]. ACM Transactions on Information and System Security, 2003, 6(3): 404-441.

[10] 徐 震, 李 斓, 冯登国. 基于角色的受限委托模型[J]. 软件学报, 2005, 16(5): 970-978.
XU Zhen, LI Lan, FENG Deng-guo. A constrained role-based delegation model[J]. Journal of Software, 2005, 16(5): 970-978.

[11] 张 宏, 贺也平, 石志国. 基于周期时间限制的自主访问控制委托模型[J]. 计算机学报, 2006, 29(8): 1427-1437.
ZHANG Hong, HE Ye-ping, SHI Zhi-guo. A delegation model for periodicity constraints-based DAC[J]. Chinese Journal of Computer, 2006, 29(8): 1427-1437.

[12] 翟征德, 冯登国, 徐 震. 细粒度的基于信任度的可控委托授权模型[J]. 软件学报, 2007, 18(8): 2002-2015.
ZAI Zheng-de, FENG Deng-guo, XU Zhen. Fine-grained controllable delegation authorization model based on trust worthiness[J]. Journal of Software, 2007, 18(8): 2002-2015.

编 辑 税 红

·我校科研成果介绍·

负信噪比信号的侦察与干扰技术研究

该项目具有以下创新点：

(1) 通过对现有检测方法的全面研究，推导了检测器的一种统一框架，并根据该统一框架提出了DSSS信号侦察与解扩的系统实现方案，完全摒弃了传统的先解调再解扩的思路。

(2) 自主提出了一种信号检测方法，该方法通过检测信号为伪码周期进行信号检测(传统方法主要通过载频检测进行信号检测)，能在很低的信噪比(–22 dB)条件下完成信号检测，并且还具有所需样本长度短、对窄带干扰和非平稳噪声有较好的抵抗能力、计算效率高等特点。

(3) 通过改进原有的利用特征分解法估计伪码形算法，提出了一种基于奇异值分解的失步点及伪码波形联合估计的快速算法，该算法具有在不影响性能的情况下计算速度很快、占用内存小、便于估计和具有较长伪码周期的伪码波形；传统的特征分解法的计算复杂度为$O(n^3)$，而该算法的计算复杂度为$o((\sqrt{n})^3)$。

(4) 引入了一种高阶循环矩方法，能在未知脉冲成形参数的情况下，有效精确地估计码元速率。