第 29 卷 第 8 期
2006 年 8 月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 29 No. 8
Aug. 2006

基于量化角色的可控委托模型

翟征德
（中国科学院软件研究所信息安全国家重点实验室 北京 100080）

摘 要 针对现有 RBAC 委托模型在支持细致委托粒度和权限传播的可控性上存在的不足，提出了量化角色的概念，实现了一种细粒度的委托约束机制，给出了一个形式化的基于量化角色的可控委托模型 QBCDM（Quantified-role Based Controllable Delegation Model）. 该模型提供了灵活的委托粒度，支持对角色任意部分权限的委托，避免了引入较高的管理代价；支持强制委托约束和细粒度的自主委托约束，保证了多步委托过程中委托能力的收敛性，显著增强了委托过程的可控性.

关键词 访问控制；基于角色的访问控制；基于角色的委托；量化角色；约束
中图法分类号 TP309

Quantified-Role Based Controllable Delegation Model

ZHAI Zheng-De
(*State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

Abstract Delegation is an important secure policy that RBAC should support. Most Existing RBAC delegation models cannot effectively support fine delegation granularity and controllable propagation of permissions. This paper first introduces the concept of quantified-role and then proposes a fine-grained delegation constraint mechanism. Lastly, it gives a formal Quantified-role Based Controllable Delegation Model (QBCDM). The model supports the delegation of any part of permissions of roles, while avoiding incurring high administrative cost. It also provides mandatory constraints and fine-grained discretionary constraints, which ensure the constringency of delegation ability in multi-step delegations and significantly enhance controllability of delegation.

Keywords access control; role based access control; role based delegation; quantified-role; constraints

1 引 言

基于角色的访问控制（Role Based Access Control, RBAC）作为一种可以有效实施组织安全策略的访问控制模型得到了广泛认可[1~4]. 它通过引入角色的概念实现了用户和权限的逻辑分离，给管理员提供了一种从组织角度进行安全建模的有效途径，显著降低了组织权限管理的负担. 近年来很多学者和技术厂商做了大量的工作以进一步增强 RBAC 的表达能力，其中基于角色的委托成为了一个研究热点[5~11].

委托是组织中常见的一种安全策略形式[12,13]. 它的主要思想是系统的主动实体能够把他的权威委托给其他主动实体，以便后者能够代表前者执行某些职能[7]. 委托有多种形式，包括用户-用户、用户-机器和机器-机器委托. 本文只关注用户-用户委托，在这种情况下，主动实施委托的实体称为委托人

收稿日期：2006-03-30；修改稿收到日期：2006-05-30. 本课题得到国家自然科学基金（60025205，60273027）资助. 翟征德，男，1979 年生，博士研究生，主要研究方向为信息与系统安全. E-mail：zhaizhengde@is.iscas.ac.cn.

(delegator),获得本次委托的权限的实体称为受托人(delegatee).以下几种情况下,组织可能需要实施委托[8].

(1)职能备份.某人出差或者长时间休假,因而需要将其工作权力委托给其他人,以保证原工作职能的继续.

(2)权力下放.当组织建立或者重组时,权力可以通过委托的方式从高级职位向低级职位扩散.

(3)工作协同.组织间或者组织内部需要协同工作时,可能需要以委托方式给参与者授权.

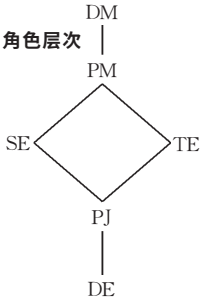
委托粒度和委托约束是 RBAC 委托机制需要特别关注的问题.在有些情况下,委托整个角色是不恰当的,安全策略要求用户只能委托角色的部分权限(部分委托).RBAC 委托机制在提供细致委托粒度的同时,应该避免引入过高的实施和管理代价,保持 RBAC 的管理优势.委托过程可能不仅包括对权限的委托,还包括委托权的委托,受托人可以将得到的权限再次委托出去,从而形成多步委托.多步委托可以显著增强系统的灵活性,然而无限制的多步委托造成了权限传播的不可预测性,给系统带来了安全隐患.因此,RBAC 委托机制需要提供一种细粒度的委托约束来有效地限制权限的传播.

本文提出了量化角色的概念,实现了一种强制与自主相结合的细粒度的委托约束机制,给出一个形式化的基于量化角色的可控委托模型 QBCDM (Quantified-role Based Controllable Delegation Model).量化角色是对普通 RBAC 角色的扩展,可以描述其任意部分权限.委托模型以量化角色作为委托单位,提供了灵活的委托粒度支持,避免了引入较高的管理代价.该模型的委托约束综合了强制和自主特征,实现了细粒度的委托限制,使得委托人可以在遵循系统强制约束的条件下严格限制受托人的委托能力,显著提高了多步委托过程中权限传播的可控性.第 2 节介绍了相关研究工作.第 3 节给出 QBCDM 模型.第 4 节对全文进行了总结并讨论了进一步研究的方向.

2 相关工作

Barka 和 Sandhu 最早讨论了 RBAC 环境下的委托.他们在文献[5]分析了基于角色委托模型的一些基本特性,包括委托的临时性、单调性、委托粒度、委托传播性等.他们在文献[6]提出的 RBDM0 (Role-Based Delegation Model Zero)是第一个基于角色的用户-用户委托模型.该模型基于核心 RBAC[14],不支持角色层次、部分委托和多步委托.文献[7]在 RBDM0 的基础上给出了 RBDM1,该模型可以支持角色层次,但仍然不支持部分委托和多步委托.文献[8]对 RBDM0 进行扩展,提出了 RDM2000(Role Based Delegation Model 2000).该模型虽然可以支持角色层次、多步委托和委托约束,但存在两个明显的不足.首先,它只支持角色级别的委托(完全委托),然而在某些情况下,部分委托是必须的.例如,图 1 给出了一家软件企业的研发部的角色层次图、用户/角色分配和权限/角色分配,部门经理可能希望将"项目进度控制"的权力委托给项目经理,而不希望把"项目验收"权力也委托给他;测试工程师可能希望将"测试项目代码"的权力委托给项目人员,而不希望把"提交测试报告"的权力也委托给他们.其次,RDM2000 的委托约束是静态和粗粒度的.该模型使用形如(r,cr,n)的约束来限制委托过程,其中 cr 是一个关于受托人当前已经分配角色情况的布尔表达式,该约束允许一个拥有角色 r 的用户可以将 r 或 r 的后代角色在不超过 n 步的情况下委托给满足条件 cr 的用户.这种约束仅仅限定了从某个用户/角色分配开始的委托的最大深度,而处于多步委托过程中间的委托人(同时也是前一步委托的受托人)完全可能希望根据受托人的确切可信度施加更为严格的委托限制,这需要一种可以由委托人实施的细粒度委托约束.例如,约束(SE,DE,4)允许软件工程师 C 委托角色 SE 及其委托权给 H,H 可以将 SE 及其委托权继续委托给 I,然而 H 现在根据 I 的可信度希望限制 I 只能委托 SE 给角色 PJ 的用户(而不是 DE 的用户)并且 I 只有单步委托 SE 的能力(而不允许 I 委托对 SE 的委托权).要表达这种动态的细粒度约束要求,仅仅使用全局强制约束是不够的,自主形式的细粒度动态委托约束是必须的,而且这种自主约束必须同时保证强制约束的满足性.

文献[9]给出了 PBDM(Permission-Based Delegation Model)委托模型.该模型支持角色委托和部分委托,其中部分委托是通过委托人创建临时性的委托角色并为之分配期望委托的权限来实现的.这种方法需要创建大量的可委托角色和临时性角色,导致模型的使用和管理严重复杂化.PBDM 提供的 can_delegate 约束同 RDM2000 中的约束类似,因此存在同样的问题.



UA

角色	用户
DM(部门经理)	A
PM(项目经理)	B
SE(软件工程师)	C，D
TE(测试工程师)	E，F
PJ(项目人员)	H，I，J
DE(部门员工)	K，L，M

PA

角色	权限
DM	P_Confirm(项目验收)
	P_Schedule(项目进度控制)
PM	P_Modify(修改项目设计)
SE	P_Code(修改项目代码)
TE	P_Test(测试项目代码)
	P_Report(提交测试报告)
PJ	P_View(查看项目文档)
DE	P_Print(使用部门打印机)

图 1　RBAC 模型示例

文献[10]给出了一种对 RBAC 系统的权限进行委托的方法，然而该方法只支持权限级别的委托，如果用户希望委托角色的部分权限，则需要对它们逐条进行委托，因此具有很高的管理和维护代价，延展性较差. 此外该方法只允许委托人限制受托人可以继续委托的步数，难以有效保证委托过程中委托能力的收敛性. 文献[11]给出了一个基于角色的受限委托模型，该模型以角色作为委托粒度，委托人在委托时可以规定受托人对委托角色的使用(Usage)限制，但没有考虑限制受托人继续委托角色的能力.

总地来说，基于角色的委托是当前 RBAC 研究的重要内容. 围绕这一点，学术界已经进行了许多卓有成效的工作，然而它们大都不支持细致委托粒度或实现部分委托的代价非常高. 此外，它们大都只考虑了粗粒度的静态强制委托约束，而不能很好地支持细粒度的动态自主委托约束. 正如 BLP 模型的强制和自主特性，强制约束是从系统全局的角度来提供安全性，自主约束对于实施细粒度的动态安全控制是十分必要的. 一个有效的 RBAC 委托模型应该在不引入高管理代价的前提下支持灵活的委托粒度和强制与自主相结合的细粒度委托约束机制.

3　QBCDM 模型

3.1　基础元素

本文所基于的 RBAC 模型与 ANSI INCITS 359-2004[14] 标准保持一致，本文使用了该标准中定义的如下集合、关系和操作符号.

定义 1(RBAC 模型).

$USERS$，$ROLES$，OPS，OBS 分别表示用户集、角色集、操作集、对象集.

$PRMS \subseteq 2^{(OPS \times OBS)}$，表示权限集.

$SESSIONS$，表示会话集.

$UA \subseteq USERS \times ROLES$，表示一个多对多的用户和角色的映射关系.

$PA \subseteq PRMS \times ROLES$，表示一个多对多的权限和角色的映射关系.

$assigned_permissions(r: ROLES) \rightarrow 2^{PRMS}$，表示一个将角色映射到其分配的权限的函数，$assigned_permissions(r) = \{p \in PRMS \mid (p, r) \in PA\}$.

$RH \subseteq ROLES \times ROLES$，表示一个称为继承的角色之间的偏序关系，记作$\geq$. 如果 $r_1 \geq r_2$，则角色 r_1 拥有 r_2 的权限并且 r_1 的用户都是 r_2 的用户.

$authorized_users(r: ROLES) \rightarrow 2^{USERS}$，表示在角色层次存在的情况下把角色映射到其授权用户，$authorized_users(r) = \{u \in USERS \mid r' \geq r, (u, r') \in UA\}$.

定义 2(角色间直接支配关系 DRH).

$DRH \subseteq ROLES \times ROLES$，表示角色间的直接支配关系. RH 是 DRH 的自反、传递闭包.

定义 3(集合 N').

集合 $N' = \{0\} \cup N$，其中 N 为自然数集.

3.2　基础 RBAC 模型的实现

RBAC 中每个角色 r 都关联到 DRH 中一组形如(r, r')和 PA 中一组形如(p, r)的元组，它们构成了 r 的权限来源，本文将这些元组的集合称为 r 的权限源.

定义 4(角色权限源函数 $psource$).

$psource(r: ROLES) \rightarrow 2^{DRH \cup PA}$，该函数将角色映射到其权限源，$psource(r) = \{(r, r') \mid (r, r') \in DRH\} \cup \{(p, r) \mid (p, r) \in PA\}$，不同角色的权限源互不相交. 符号$|r|$表示 r 的权限源的基，即$|r| = |psource(r)|$.

RBAC 的实现为角色 r 的权限源中的每个元组分配一个 $2^i (i \in N')$，该数值称为元组的量值. 同一角色的权限源中元组的量值各不相同，它们分别是 $2^0, 2^1, 2^2, \cdots, 2^{|r|-1}$.

定义 5(元组的量值函数 $quantity()$).

$quantity(tuple: DRH \cup PA) \rightarrow N$，该函数将 DRH 或 PA 中的元组映射到其量值.

表 1 给出了图 1RBAC 模型示例中 RBAC 角色相关元组的量值. 元组的量值对于管理员和用户都是可见的. 这里并不要求修改 RBAC 模型的关系，只要求模型的实现为每个元组维护此量值.

表 1 元组的量值

角色-角色	量值	角色-权限	量值
(DM,PM)	2^0	(DM,P_Confirm)	2^1
(PM,TE)	2^0	(DM,P_Schedule)	2^2
(PM,SE)	2^1	(PM,P_Modify)	2^2
(SE,PJ)	2^0	(SE,P_Code)	2^1
(TE,PJ)	2^0	(TE,P_Test)	2^1
(PJ,DE)	2^0	(TE,P_Report)	2^2
		(PJ,P_View)	2^1
		(DE,P_Print)	2^0

定义 6(角色的总量、总量函数 $total()$).

角色的总量是角色权限源中元组的量值的和.

$total(r:ROLES)\rightarrow N$，该函数将角色映射到它的总量，$total(r)=\sum_{tup\in psource(r)} quantity(tup)$.

3.3 量化角色

量化角色是通过对普通角色进行扩展产生的、只具有其部分权限的角色，代表由角色的权限源中的部分元组产生的权限的集合，这些元组称为量化角色的权限源. 量化角色的形式是(r,k)，其中 $r\in ROLES, k\in N\wedge 0<k\leq total(r)$是该量化角色的量.

每个角色的权限源中元组的量值各不相同，并且形成一个超递增数列，因此对任意给定的(r,k)，方程 $k=\sum_{0\leq i<|r|} a_i 2^i$ 在$\{0,1\}$上有唯一解. 如果在该方程的解集中，$a_{j_1},a_{j_2},\cdots,a_{j_m}$ $(0<m\leq|r|\wedge 0\leq j_i<|r|)$的值为 1，则集合$\{2^{j_1},2^{j_2},\cdots,2^{j_m}\}$是$(r,k)$的量值集. 如果 2^i 属于(r,k)的量值集，则(r,k)的权限源包含角色 r 的权限源中量值为 2^i 的元组. 例如按照表 1，(PM,5)的权限源为{(PM,TE),(PM,P_Modify)}.

在实际实现时，量化角色的量和元组的量值都以二进制形式保存，计算(r,k)的权限源时，如果 k 的二进制表示的第 i 位为 1，则(r,k)的权限源将包括 r 的权限源中量值为 2^{i-1} 的元组. 委托过程中只使用量化角色.

定义 7(量化角色的权限源函数).

$QRS=\{(r,k)\mid r\in ROLES\wedge 0<k\leq total(r)\}$，是所有量化角色的集合.

$kseq((r,k):QRS)\rightarrow 2^{N'}$，该函数将$(r,k)$映射到其量值集，$kseq((r,k))=\{2^i\mid k$ 的二进制表示中第 $i+1$位为 1$\}$.

$qrpsource(qr:QRS)\rightarrow 2^{UA\cup PA}$，该函数将量化角色映射为其权限源.

$$qrpsource((r,k))=(\{(r,r')\mid(r,r')\in DRH\wedge quantity((r,r'))\in kseq((r,k))\}\cup\{(p,r)\mid(p,r)\in PA\wedge quantity((p,r))\in kseq((r,k))\}).$$

定义 8(量化角色上的关系=和≥).

$$(r,k)=(r',k')\Leftrightarrow r=r'\wedge kseq(r,k)=kseq(r',k')$$

$$(r,k)\geq(r',k')\Leftrightarrow \begin{cases} kseq(r,k)\supseteq kseq(r',k'),\ r=r' \\ \exists x:ROLES\cdot(r,x)\in qrpsource(r,k)\wedge x\geq r', r\neq r' \end{cases}.$$

如果$(r,k)\geq(r',k')$，一个被明确授权委托(r,k)的用户可以委托(r',k').

定理 1. 量化角色上的≥关系是偏序关系.

证明. 现在证明≥是自反的、反对称的和传递的.

自反性：根据定义，$(r,k)\geq(r,k)$是显然的.

反对称性：

$$(r,k)\geq(r',k')\Leftrightarrow \begin{cases} kseq(r,k)\supseteq kseq(r',k'),\ r=r' & (1) \\ \exists x:ROLES\cdot(r,x)\in qrpsource(r,k)\wedge x\geq r', \\ \quad r\neq r' & (2) \end{cases}$$

$$(r',k')\geq(r,k)\Leftrightarrow \begin{cases} kseq(r',k')\supseteq kseq(r,k),\ r'=r & (3) \\ \exists x':ROLES\cdot(r',x')\in qrpsource(r',k')\wedge x'\geq r, \\ \quad r'\neq r & (4) \end{cases}$$

若式(1),(3)成立，则有$(r,k)=(r',k')$.

若式(2),(4)成立，根据式(2)，$r\geq x\geq r'$并且 $r\neq r'$，则 $r>r'$.

根据式(4)，$r'\geq x'\geq r$ 并且 $r'\neq r$，则 $r'>r$.

产生矛盾，所以式(2)和式(4)不可能同时成立.

显然式(1),(4)不可能同时成立，式(2),(3)不可能同时成立.

总之，$(r,k)\geq(r',k')\wedge(r',k')\geq(r,k)\Rightarrow(r,k)=(r',k')$.

传递性：

$$(r,k)\geq(r',k')\Leftrightarrow \begin{cases} kseq(r,k)\supseteq kseq(r',k'),\ r=r' & (5) \\ \exists x:ROLES\cdot(r,x)\in qrpsource(r,k)\wedge x\geq r', \\ \quad r\neq r' & (6) \end{cases}$$

$$(r',k')\geq(r'',k'')\Leftrightarrow \begin{cases} kseq(r',k')\supseteq kseq(r'',k''),\ r'=r'' & (7) \\ \exists x':ROLES\cdot(r',x')\in qrpsource(r',k')\wedge x'\geq r'', \\ \quad r'\neq r'' & (8) \end{cases}$$

若式(5),(7)成立，则 $r=r''$且 $kseq(r,k)\supseteq kseq(r'',k'')$，即$(r,k)\geq(r'',k'')$.

若式(5),(8)成立，则由(8)，$r'\geq x'\geq r''$且 $r'\neq r''$，结合式(5)，则 $r=r'>r''$.

则由$(r',x')\in qrsource((r',k'))$，得到$(r,x')\in qrsource(r,k)$，结合$x'\geq r''$，则有$(r,k)\geq(r'',k'')$.

若式(6)，(7)成立，则$r\neq r''$并且$x\geq r''$，则有$(r,k)\geq(r'',k'')$.

若式(6)，(8)成立，则有$r>r'>r''$，$x\geq r'>r''$，则$(r,k)\geq(r'',k'')$.　　证毕.

量化角色在$\geq$下形成角色层次，该角色层次依赖于RBAC模型中的角色层次和PA关系. 量化角色是对RBAC角色的扩展，可以表示其任意部分权限，使用量化角色作为委托单位提供了灵活的委托粒度，同时避免了引入较高的管理和维护代价.

3.4　委托关系

一个用户向其他用户进行委托时，系统产生委托票据，该委托票据的内容被记录在委托关系中，作为用户行使权力或进一步委托的依据. 实现细粒度的委托约束需要区分对量化角色qr的行使权和委托权. 本文假定委托人只有在同时拥有qr的行使权和委托权的情况下，才能委托该量化角色. 用户可以只拥有量化角色的行使权而不拥有委托权，但相反的情况是不允许的. 此外，基于本次委托(委托qr)进行的所有后继委托不能再将qr委托给本次委托的委托人，以避免在委托过程中出现回路. 在委托qr时，委托人不仅可以限制受托人能够继续委托该量化角色的步数还可以限制受托人能够将它进一步委托给哪些用户. 为了保证多步委托过程中委托能力的收敛性，本次委托的约束在所有基于本次委托而进行的后继委托中也必须得以满足.

定义9(委托条件).

一个委托条件$dc\in DC$是一个使用"$\wedge$"(逻辑与)连接r、$\bar{r}$形式的原子公式构成的布尔表达式，其中r要求用户已经被分配了该角色，$\bar{r}$的含义与之相反.

集合$ATOMS=\{r,\bar{r}\mid r\in ROLES\}$，是原子公式的集合.

$satisfy(dc:DC)\rightarrow 2^{USERS}$，一个将委托条件映射到满足该条件的用户的函数.

$fatoms(dc:DC)\rightarrow 2^{ATOMS}$，一个将委托条件映射到其使用的原子公式的函数.

委托条件dc必须是自身不包含矛盾的，即$\neg(\exists r_1:ROLES\cdot\exists r_2:ROLES\cdot r_1\in fatoms(dc)\wedge\bar{r}_2\in fatoms(dc)\wedge r_1\geq r_2)$.

委托条件$dc'\in DC$可以蕴含$dc\in DC$，$(dc'\rightarrow dc)\Leftrightarrow(\forall r_1:ROLES\cdot r_1\in fatoms(dc)\Rightarrow\exists r_2:ROLES\cdot r_2\in fatoms(dc')\wedge r_2\geq r_1\wedge\forall r_1:ROLES\cdot\bar{r}_1\in fatoms(dc)\Rightarrow\exists r_2:ROLES\cdot\bar{r}_2\in fatoms(dc')\wedge r_1\geq r_2)$.

定义10(委托请求、委托票据、委托关系).

委托请求是一个四元组$(de,qr,steps,dc)$，其中$de\in USERS$，代表受托人；$qr\in QRS$，是本次委托的量化角色；$steps\in N'$，表明de可以继续委托qr的步数；$dc\in DC$，代表de继续委托qr时受托人应该满足的条件，当$steps=0$时，受托人只拥有对qr的使用权，而没有委托权，此时可以忽略条件dc. $steps$和dc是委托人自主施加的约束，限制了受托人继续委托qr的能力.

$DRS\subseteq USERS\times QRS\times N'\times DC$，是委托请求的集合.

委托票据是一个由用户发起的、经过系统授权的委托请求，它是一个二元组(dr,drs)，其中$dr\in USERS$，代表委托人；$drs\in DRS$，代表委托请求.

委托关系DT是委托票据的集合，$DT\subseteq USERS\times DRS$.

3.5　委托授权

目前为止，委托人使用委托票据可以有效地限制受托人的委托能力. 然而，直接分配了普通角色的用户可以任意地将他所拥有的权限委托给其他用户，这在某些情况下是不恰当的. 例如，在图1RBAC模型示例的例子中，企业可能希望部门经理只能委托"项目进度控制"的权力给项目经理，而不希望他委托"项目验收"的权力给其他人. 事实上，分配了普通角色的用户都可以作为一个有向非循环委托图的起点，对这些用户的委托能力施加的约束限制了委托图中委托路径的最大深度和所有受托用户需要满足的条件. 可委托关系CAN_D是由系统管理员制定和维护的强制约束.

定义11(可委托关系CAN_D).

$CAN_D\subseteq ROLES\times QRS\times N\times DC$，满足$\forall(r,(r',k'),n,dc):CAN_D\cdot(r,total(r))\geq(r',k')$. 若$(r,(r',k'),n,dc)\in CAN_D$，则拥有普通角色$r$(或$r$的祖先角色)的用户可以在受托人满足条件$dc$并且委托步不超过$n$的情况下委托$(r',k')$.

当委托人希望向受托人委托时，他提交一个委托请求drs. 系统根据委托关系表DT或可委托关系CAN_D决定是否授权本次委托，如果系统判定drs是合法的，则接受该请求并在DT产生一个相应的委托票据，本次委托生效；否则拒绝接受drs.

定义12(委托请求的接受条件).

由用户dr发起的委托请求$(de,qr,steps,dc)$在满足下列条件时可以被系统接受：

(1) $\exists(r',qr',n',dc'):CAN_D\cdot(\exists r:ROLES\cdot(r\geq r'\wedge(dr,r)\in UA)\wedge qr'\geq qr\wedge n'>steps\wedge dc\rightarrow dc'\wedge$

$de \in satisfy(dc')$)，即根据 CAN_D，受托人 de 满足 dc' 条件，委托人 dr 拥有 qr 并且其对 qr 的委托能力大于本次委托中要委托给受托人的委托能力.

或者

(2) $\exists(sbody,(dr,qr,n'',dc''))$：$DT \cdot de \in satisfy(dc'') \wedge n'' > steps \wedge dc \rightarrow dc''$，即根据 DT，受托人 de 满足 dc'' 条件，委托人 dr 拥有 qr 并且其对 qr 的委托能力大于要委托给受托人的委托能力.

在多步委托过程中，委托能力是逐步收敛的. CAN_D 构成了委托系统的强制约束，它不仅限制分配了普通角色的用户的委托能力，也对所有的后继委托施加了限制. 在后继委托过程中，委托人可以在满足 CAN_D 约束的情况下自主地施加更为严格的委托限制. 这显著增强了委托过程的可控性.

3.6 撤 销

根据文献[15]，委托撤销可以分为强(strong)和弱(weak)撤销，也可以分为层叠(cascading)和非层叠(noncascading)撤销. 根据文献[6]，委托撤销又可以分为授予依赖(grant-dependent)和非授予依赖(grant-independent)撤销. 这些撤销模式是彼此正交的，可以组合产生 8 种具体撤销模式.

本模型支持所有这些撤销模式. 在撤销某个对量化角色(r,k)的委托时，可以要求撤销人是本次委托的委托人(授予依赖)或者是角色 $r'(r' \geq r)$ 的用户(非授予依赖)，可以仅撤销受托人的(r,k)(弱撤销)或者撤销受托人被委托的所有$(r',k')((r',k') \geq (r,k))$(强撤销)，可以只撤销本次委托的票据(非层叠撤销)或者撤销所有基于本次委托而进行的后继委托(层叠撤销).

3.7 模型的结构

QBCDM 模型的结构如图 2 所示. 该模型在原有 RBAC 模型的基础上引入了量化角色，量化角色之间形成角色层次，该角色层次依赖于原 RBAC 模型中的角色层次和角色/权限分配关系. 此外该模型还引入了 CAN_D 强制约束和委托关系中的自主约束.

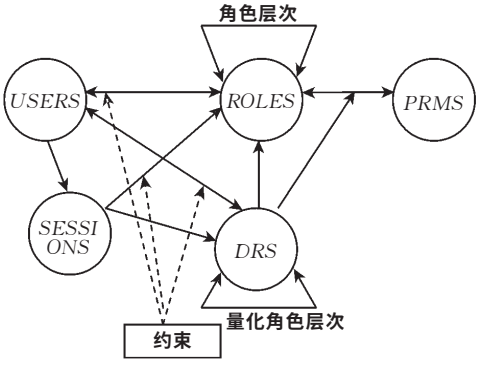


图 2 QBCDM 结构

3.8 模型的应用

下面举例说明模型的应用. 假设图 1 中的软件企业希望在研发部实施如下委托策略：

(1) 部门经理可以将"项目进度控制"的权限委托给项目经理，项目经理不能再次委托该权限.

(2) 测试工程师可以将角色 PJ 和"测试项目代码"的权限委托给部门员工，而且允许部门员工再次委托该权限，但要求委托深度不超过 3.

则系统安全管理员可以制定如下强制委托约束 c_1 和 c_2：

c_1. (DM,(DM,4),1,PM)

c_2. (TE,(TE,3),3,DE)

现在测试工程师 E 要把(TE,3)和其委托权委托给员工 K，但希望同时实施如下约束：

(1) K 可以委托该量化角色的行使权，但不能再委托对其的委托权.

(2) K 不能将(TE,3)委托给软件工程师.

E 产生委托请求(K,(TE,3),1,DE $\wedge \overline{\text{SE}}$)，根据 c_2 该请求被系统接受，票据 t=(E,(K,(TE,3),1,DE $\wedge \overline{\text{SE}}$))被加入到 DT 中.

现在 K 希望将向员工 C 委托(TE,3)的行使权. 他产生委托请求(C,(TE,3),0,DE $\wedge \overline{\text{SE}}$)，该请求会因为 C 不能满足 t 中的条件 DE $\wedge \overline{\text{SE}}$ 而被拒绝.

4 结 论

针对现有 RBAC 委托模型在支持细致委托粒度上的不足，本文引入了量化角色的概念并将其应用于委托. 量化角色是普通 RBAC 角色的扩展形式，可以描述 RBAC 角色的任意部分权限，以较低的管理和维护代价实现了灵活的委托粒度. 为了增强委托过程中权限传播的可控性，该模型实现了一种综合了强制和自主特征的委托约束，实现了细粒度的委托限制，保证了多步委托过程中委托能力的收敛性，提高了多步委托过程中权限传播的可控性. 本文形式化地定义了量化角色和基于量化角色的可控委托模型(QBCDM).

量化角色是一种在 RBAC 系统中实施最小特权原则的有效机制，委托是它的一种具体应用形式. 研究量化角色在其它 RBAC 场景中的应用是我们感兴趣的问题. 基于量化角色的可控委托模型的职责分离约束是我们正在考虑解决的问题.

致 谢 在此，我们向本文的匿名评审专家表示感

谢，感谢他们对本文的所提出的宝贵意见和建议；对于本实验室的徐震博士的热情帮助，也一并致谢！

参 考 文 献

1 Ferriaolo D., Cugini J., Kuhn R.. Role-based access control (RBAC): Features and motivations. In: Proceedings of the 11th Annual Computer Security Application Conference, New Orleans, 1995, 241～248

2 Sandhu R. S., Coyne E. J., Feinstein H. L., Youman C. E.. Role-based access control models. Computer, 1996, 29(2): 38～47

3 Ferraiolo D., Kuhn R.. Role-based access control. In: Proceedings of the 15th National Computer Security Conference, 1992, 554～563

4 Kern A., Schaad A., Moffett J.. An administration concept for the enterprise role based access control mode. In: Proceedings of the 8th ACM Symposium on Access Control Models and Technologies, New York, 2003, 3～11

5 Barka E., Sandhu R.. Framework for role-based delegation models. In: Proceedings of the 16th Annual Computer Security Application Conference, 2000, 168～176

6 Barka E., Sandhu R.. A role-based delegation model and some extensions. In: Proceedings of the 23rd National Information Systems Security Conference (NISSC 2000), Baltimore, 2000

7 Barka E., Sandhu R.. Role-based delegation model/hierarchical roles (RBDM1). In: Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04), 2004, 396～404

8 Zhang L. H., Ahn G-J, Chu B-T. A rule-based framework for role-based delegation. In: Proceedings of the 6th ACM Symposium on Access Control Models and Technologies, New York, 2001, 153～162

9 Zhang X. W., Oh S., Sandhu R. S.. PBDM: A flexible delegation model in RBAC. In: Proceedings of the 8th ACM Symposium on Access Control Models and Technologies, New York, 2003, 149～157

10 Wainer J., Kumar A.. A fine-grained, controllable user-to-user delegation method in RBAC. In: Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, Stockholm, 2005, 59～66

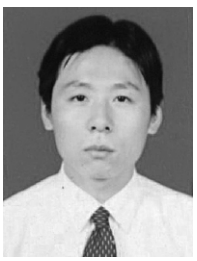
11 Xu Zhen, Li Lan, Feng Deng-Guo. A constrained role-based delegation model. Journal of Software, 2005, 16(5): 970～978 (in Chinese)
(徐　震，李　斓，冯登国. 基于角色的受限委托模型. 软件学报, 2005, 16(5): 970～978)

12 Gasser M., McDermott E.. An architecture for practical delegation in a distributed system. In: Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, 1990, 20～30

13 Bandmann O., Dam M., Firozabadi B. S.. Constrained delegation. In: Proceedings of the 23rd Annual IEEE Symposium on Security and Privacy, Oakland, 2002, 131～143

14 ANSI INCITS 359-2004. Role based access control. American national standard for information technology, 2004

15 Hagstrom A., Jajodia S., Presicce F. P., Wijesekera D.. Revocations-a classification. In: Proceedings of the 14th IEEE Computer Security Foundations Workshop, Nova Scotia, Canada, 2001, 44～58



ZHAI Zheng-De, born in 1979, Ph. D. candidate. His research interests include information and system security.

Background

The work is supported by the National Natural Science Foundation of China under grant No. 60025205 and 60273027.

Delegation in Role Based Access Control Systems provides an effective means for decentralization of administrative task. However, most existing role-based delegation models either only support delegation at the whole role level and does not support partial delegation, which is rather needed in many situations, or enforce partial delegation in a very costly way, which often includes creating many new temporary roles, changing the original role hierarchies. Furthermore, they often only consider coarse-grained static mandatory delegation constraints, which can not effectively control the propagation of permissions in multi-step delegations. Multi-step delegation without strict constraint on permission propagation can be a serious threat to model's safety.

This paper first introduces the concept of quantified role, which is an extension of regular role and can represent any part of the role's permissions, and then propose a fine-grained delegation constraint mechanism. Lastly, it gives a Quantified-role Based Controllable Delegation Model (QBCDM). The model can supports flexible delegation granularity in an efficient way and enable the grantor to strictly control the delegatee's power to further delegate the permissions he has been delegated, which ensures the constringency of delegation ability in multi-step delegations and significantly enhances the controllability of delegation.