基于时限的角色访问控制委托模型*)

道 炜[1,2] 汤 庸[1] 冀高峰[1] 杨虹轶[1]

(广州中山大学信科院协同软件实验室 广州 510275)[1] (广东天讯电信科技有限公司 广州 510620)[2]

摘 要 访问权限的委托限制是一种重要的安全策略,它的基本思想是用户将自己所具有的部分或者全部权限转授给其他用户,让接受授权的用户代表发出授权的用户执行某些任务。基于角色的委托授权模型将角色作为委托的主体。目前为止它的两个基本模型 RBDM0 和 RDM2000 都没有有效地解决时限问题,事实上时限是授权的重要组成部分,本文在上述两个模型的基础上引入有效时间和角色激活的概念,描述了与之相对应的带时限的委托模型并给出了相应的委托判断规则。

关键词 访问控制,委托授权,时限,有效时间

Delegation Model for Timing Constraints-based RBAC

DAO Wei[1,2] TANG Yong[1] JI Gao-Feng[1] YANG Hong-Yi[1]

(Computer Science Department of Sun Yat-sen University, Guangzhou 510275)[1] (Tianxun Telecom Ltd,Guangdong, Guangzhou 510620)[2]

Abstract Delegation is an important security policy, the basic idea is user give its own permissionto another user, then the delegated user use those permission to execute some tasks. The basic RBAC-delegation model RBDM0 and RDM2000 can't solve temporal constraints, actually temporal constraint is an important part of authorization. In this paper, We present a delegation model to support temporal constraint ,and then describe delegation rule.

Keywords Access control, Delegation authorization, RBAC, Timing constraint

1 引言

访问控制技术的研究一直是信息安全研究的热点问题。访问控制技术起源于 20 世纪 70 年代,在随后的三十多年中,先后出现了多种访问控制模型:自主访问控制模型,强制访问控制模型,基于角色的访问控制模型 RBAC。

访问权限委托限制是一种重要的安全策略。它的基本思想是用户将自己所具有的部分或者全部权限转授给其他用户,让接受授权的用户代表发出授权的用户执行某些任务。从委托的方式[1,2]看主要有以下几种:1)永久委托/临时委托:如果委托用户在转授出角色之后,再也收不回来则是永久委托;如果委托的过程只是临时的,则是临时委托。2) 单调委托/非单调委托:如果委托用户在转授出自己拥有的角色之后,仍然享有该委托角色的权限,则是单调委托。如果委托之后,失去了该委托角色的权限直到该委托关系失效,则是非单调委托。3)完全委托/部分委托:如果委托角色的用户将该角色的全部权限都转授给被委托的用户,则称为完全委托。如果委托的只是角色的部分权限,则称为部分委托。

本文将在基于角色的访问控制模型之下讨论角色到角色的委托访问权限限制,对角色委托的临时性、时序依赖性和受限传播性这些约束特性进行形式化建模。第 2 节介绍目前访问控制委托模型的相关研究工作,第 3 节阐述基于角色的带时限的访问控制委托模型及其相关委托判断规则,最后总结全文并指出进一步的研究方向。

2 相关工作

基于角色的访问控制技术普遍被认为是一种很有前途的全新的安全信息技术,具有代表性的是 Sandu 等提出的 RBAC96[3]模型,包括五种基本元素:用户、角色、权限、会话、约束。两种关系:用户指派关系(UA)、权限指派关系(PA)。

基于角色的委托模型主要有 RBDM0 和 RDM2000。基于 RBAC96,RBDM0[4] 首次将角色的概念引入到委托模型中,它将用户的角色分为两类初始角色和委托角色。

RBDM0 的概念:

$UAO \subseteq U \times R$ 是用户与初始角色之间的多对多关系;

$UAD \subseteq U \times R$ 是用户与委托角色之间的多对多关系;

$UA = UAO \cup UAD$

$Users: R \rightarrow 2^U$ 是一个角色到一组用户的映射函数,$Users(r) = \{u \mid (u,r) \in UA\}$

$Users(r) = Users\_O(r) \cup Users\_D(r)$

$Users\_O(r) = \{u \mid (\exists r' \geqslant r)(u,r' \in UAO)\}$

$Users\_D(r) = \{u \mid (\exists r' \geqslant r)(u,r' \in UAD)\}$

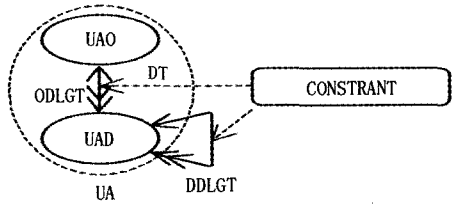

图 1 RDM2000 委托模型

Longhua Zhang 等人提出的 RDM2000[5]委托模型如图 1 支持层次角色和多步委托,是基于 RBDM0 扩展而来的相对完善的基于角色的委托模型。

以上这些模型都没有解决时限问题,事实上时限是授权

*)基金项目:国家自然科学基金项目(60673135,60373081)、广东省自然科学基金项目(04105503,5003348)、教育部"新世纪优秀人才支持计划"资助项目。

的重要组成部分[6]，例如某用户出差为了继续完成他的工作，需要将他承担的角色委托给他人，他返回时则需要回收权限，又如某用户被委托的角色在某一时段或时点是否有效依赖于在这一时段或时点角色的委托是否有效，这都说明角色委托具有时效性和时序性。孙波[7]等人在委托模型中引入了时间的概念，但他不支持委托角色中的部分权限。本文将总结带时限的委托模型的特点，不仅支持带时限的完全委托授权，也支持委托转授角色中的部分权限。

3 带时限的委托模型

3.1 时间的相关定义

定义 1 时间区间：时间区间是由两个时间点所构成的区间即

$TI=\{(p_i,p_j)\mid p_i,p_j\in TP,p_i\leqslant p_j\}$

定义 2 时间区间的集合：

$TIS=2^{TI}$，表示由时间区间所组成的集合。

定义 3 两个时间区间之间的包含关系：对于 $\forall t_1,t_2\in TI$，若 t_1 与 t_2 满足 Allen 提出的十三种时态区间关系[8]的以下四种关系的任意一种：

(1)$During(t_1,t_2)$，即 t_1 比 t_2 晚开始，且早结束，在时间轴 t_1 上的区间范围被包含在 t_2 的区间范围内；

(2)$Starts(t_1,t_2)$，即 t_1 和 t_2 有共同的起始点，但 t_1 比 t_2 先结束；

(3)$Finishes(t_1,t_2)$，即 t_1 和 t_2 有共同的结束点，但 t_1 比 t_2 晚，开始；

(4)$Equals(t_1,t_2)$，即 t_1 和 t_2 有共同的时间区间，t_1 和 t_2 在时间轴上重合；

定义 4 时间区间集合之间的属于关系：对于 $\forall t\in TI$，$S_t\in TIS$，如果 $\exists t'\in S_t$ 使得 $t\subseteq_t t'$，则称时间区间 t 属于时间区间的集合 S_t，记为 $t\in_t S_t$。

定义 5 时间点与时间区间集合的属于关系：对于 $\forall p\in TP,S_t\in TIS$，如果 $\exists t\in TI$，使得 $p\in_p t$ 且 $t\in_t S_t$ 成立，则称时间点 p 属于时间区间的集合 S_t，记为 $p\in_s S_t$。

定义 6 两个时间区间集合之间的包含关系：如果两个时间区间的集合 $S_t,S'_t\in TIS$，满足对于 $\forall t\in S_t$，均有 $t\in_t S_t$，则称时间区间的集合 S_t 包含在 S'_t 中，记为 $S_t\subseteq_s S'_t$。

3.2 带时限的委托模型基本定义

考虑到用户在系统中对客体对象的访问是通过用户—角色—权限的层层映射关系来实现的，如果为用户到角色的映射关系 UA 添加有效时间的控制，也就为用户的访问控制添加了有效时间，这是一种动态的授权方式，用户只有在有效的时间内才具备有效的角色，才具备有效的权限。

在 RBAC96 模型中增加具有有效时间的用户角色指派关系定义。

定义 7 具有有效时间的用户角色指派关系

$UAT\subseteq U\times R\times TIS,UAT=\{(u,r,S_t)\mid(u,r)\in UA,S_t\in TIS\}$

$UAOT\subseteq U\times R\times TIS,UADT\subseteq U\times R\times TIS,UAT=UAOT\cup UADT$

定义 8 具有有效时间的委托关系：

$DLGTT\subseteq UAT\times UAT$

即$((u,r,S_t),(u',r',S'_t))\in DLGTT$ 表示在有效时间 S_t 内具有角色 r 的用户 u，把角色 r' 转授给用户 u'，并且在 S'_t 的范围内有效。其中，角色 r' 是等同于 r 的角色或者是比 r 低级的角色，并且有效时间集合 S'_t 必须是包含在 S_t 中的。

定义 9 多步委托中具有有效时间：

具有有效时间的委托路径：$DPT\subseteq UAT\times UAT$

具有有效时间的委托树：$DTT\subseteq UAT\times UAT$

从定义中可以看出 DTT 表示树的方法是用树中的边的关系来表示委托权限之间的关系。那么树中的节点 $Node$ 必须满足：

$DTT\subseteq Node\times Node\subseteq DLGTT\subseteq UAT\times UAT$

3.4 加入有效时间后的委托模型

本小节首先对加入有效时间控制之后的委托模型和反应角色继承关系的委托树进行了描述。

·委托模型的有效时间扩展：

采用$([t_b,t_e],p)$来表示委托权限的有效期限和使用时间约束，$[t_b,t_e]$ 是一个时间段 duration，t_b 是时间段 duration 的开始，t_e 是时间段 duration 的结束，其中 $t_b\in N,t_e\in N,t_b<t_e$。$P$ 是时间周期表达式。具有时限的转授权的具体含义是委托用户在转授权的操作中，仅仅赋予被委托用户在时间段 duration 以及时间周期 P 中具有授予的角色 r，即被委托用户仅仅在有效时间内可以行使授予的角色 r 所具有的权限，一旦当前时间 $t>t_e$ 或不在 P 范围中，则系统自动撤销被委托用户拥有授予的角色 r。例$(r,([2007/2/2,\ 2007/5/1],weekend))$表示用户只有在 2007 年 2 月 2 日至 2007 年 5 月 1 日的周末才能将角色 r 委托出去。

·委托模型的类型

(1)在有效时间范围内，基于角色的完全委托。即转授的权限都是以整体角色为单位，转授的权限是某个角色的全部权限。

(2)在有效时间范围内，基于角色的部分委托。即转授的权限仅仅是某个角色的部分权限，但是通过将部分权限授予给一个临时的角色来转授给用户，转授权的单位仍是角色，可参考文[9]的处理方式。

(3)对于已有的转授权关系$((u,r,S_t),(u',r',S'_t))\in DLGTT$，增加$(u',r',S'_t)$的有效时间。即对 S'_t 作更新操作。

·有效时间对委托树结构的影响

委托实际是一种角色转授、继承关系的反应，委托树则是将这种关系用树状的形式表达出来。图 2 给出了角色层次关系，表 1 是对有效时间加入用户角色对应关系的例子描述。

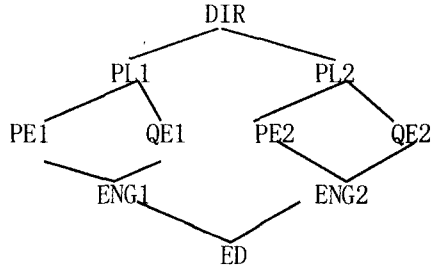

图 2 角色层次关系例子

表 1 用户-角色-有效时间集合的示例

用户名	角色名	有效时间
Mike	DIR	{[1,10],[20,30], ……}
John	PL2	{[1, 20],[40, 50],……}
Betty	QE1	{[1,30],[60, 70], ……}
Tom	PE2	{[1, 5], [10, 25], ……}
Bob	ENG1	{[2,10], [45, 90], ……}
Cathy	ED	{[1,30],[35, 55], ……}

对上例我们分析一下有效时间为委托授权带来的影响

1. 有效时间范围内基于完整角色的委托授权：

$((Mike,DIR,[1,10]),(John,DIR,[2,9]))\in DLGTT$

$((Mike,DIR,[1,10]),(Betty,PL1,[2,7]))\in DLGTT$

$((Mike,DIR,[1,10]),(Betty,DIR,[5,10]))\in DLGTT$

$((Betty,PL1,[2,7]),(Cathy,QE1,[3,4]))\in DLGTT$

$((Betty,PL1,[2,7]),(Bob,PE1,[2,5]))\in DLGTT$

$((Betty,DIR,[5,10]),(Tom,PE2,[6,8]))\in DLGTT$

2. 在有效时间范围内，基于角色的部分委托授权：

$((John,DIR,[2,9]),(Tom,\{PL2,p\_range\},[2,9]))\in DLGTT$

这个部分转授权的例子，为了清楚地显示 Tom 被授的角色的权限，用了$\{PL2,p_range\}$的表达方式，实际在处理的时候，是将$\{PL2,p_range\}$的权限赋予了一个临时的角色 T_R，再将 T_R 和用户 Tom 建立对应关系即可。被授予角色中的部分权限的用户不能再将这个临时的角色 T_R 转授给他人，这是出于安全性的考虑。

通过上述完全和部分委托授权的操作，就形成了图 3 中的带有有效时间的委托授权树。

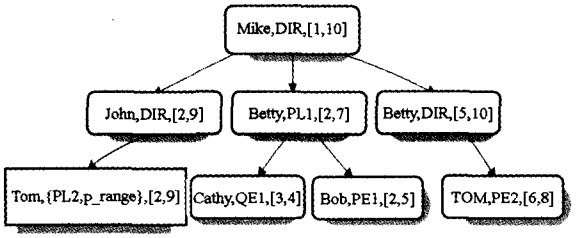

图 3　带有有效时间的委托授权树

3. 已存在的委托授权关系，增加授权关系的有效时间：

(1)情形一、有效时间不会对委托授权树的结构造成影响，例：已有的转授权关系

$((Betty,DIR,[5,10]),(Tom,PE2,[6,8]))\in DLGTT$

$(Betty,DIR,[5,10])$ 在新增 $((Betty,DIR,[5,10]),(Tom,PE2,[8,9]))\in DLGTT$ 之后，则需要对有效时间的区间进行合并。合并之后的关系$((Betty,DIR,[5,10]),(Tom,PE2,[6,9]))\in DLGTT$，转授权树的结构不会受到影响，只是需要将树中原有的结点$(Tom,PE2,[6,8])$修改为$(Tom,PE2,[6,9])$。

(2)情形二、有效时间会对委托授权树的结构造成影响，例：如果$((Mike,DIR,[1,10]),(Cathy,QE1,[3,8]))\in DLGTT$ 成立，那么原有的树的结构就要发生变化，这是由于$(Cathy,QE1,[3,4])$被修改了有效时间导致。因为原有的关系$((Betty,PL1,[2,7]),(Cathy,QE1,[3,4]))G\in DLTT$ 在发生$(Cathy,QE1,[3,4])\rightarrow(Cathy,QE1,[3,8])$的变化之后，已经超过了$(Betty,PL1,[2,7])$的有效时间。则新的委托树如图 4 所示。

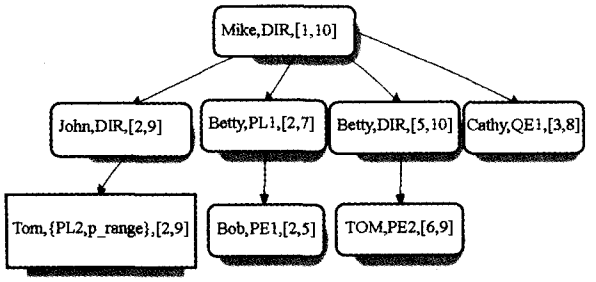

图 4　增加有效时间引起委托树的变化

以上变化说明有效时间改变了原有的转授权关系。

3.5　带时限的委托判断规则

RDM2000 定义了允许常规用户委托转授其角色的策略，同时也描述了如何撤销这些转授出去的角色的策略，使用基于规则的策略描述语言(rule-based policy specification language)可以使这些策略有效地执行。

语句(Clause)，也称作规则(Rule)，表达形式为：$H\leftarrow B$。其中，H 代表规则头(rule head)，B 代表规则体(rule body)。

若 B 为真则将触发 H 为真，这正好提供描述授权和执行授权的机制。因此，授权策略需要满足的条件可以置于规则体内，而授权本身则置于规则头的位置。

规则 3-1　带时限用户-用户的完全转授权判定规则

$total\_can\_delegate(u,r,u',r',S'_t,dlg\_opt)\leftarrow$
$active(u,r,s)\&$
$further\_delegatable(u',r',dlg\_opt)\&$
$senior(r,r'')\&$
$junior(r',r'')\&$
$can\_delegate(r'',cr,d,w)\&$
$has\_relation(u',cr)\&$
$lt(depth(u,r,valid\_TIS(u,r)),d)\&$
$lt(width(u,r,valid\_TIS(u,r)),w)\&$
$in(S'_t,valid\_TIS(u,r)).$

其中，u、r、u'、r'、S'_t 分别表示委托转授用户、委托转授角色、被委托授权用户、被委托授权角色、u' 与 r' 对应关系的有效时间集合。dlg_opt 则表示是否允许下一步转授权。若 dlg_opt 为 true，则 $further\_delegatable(u',r',dlg\_opt)$ 为 true，否则为 false。

该规则表达的意思是，在会话 s 中激活角色 r 的用户 u 能将角色 r' 授予给用户 u'，有效时间集合为 S'_t，并指明是否允许被授用户进行下一步转授权，但必须满足以下所有的条件：r 比 r' 高级；不超过最大转授权深度和宽度；u' 必须满足前置条件 cr；S'_t 必须在 u 与 r 对应关系的有效时间集合的范围之内。

注意，由定义 5 的时间点属于时间区间集合的定义，本文认为，如果对于 $\forall p_t\in{}_sS'_t\ \exists\ conflicting(rolesT(u',p_t),r')$ 为 true，即两角色存在冲突，则委托授权操作将返回失败。

规则 3-2　带时限用户-用户的部分转授权判定规则

$partial\_can\_delegate(u,r,u',r',p\_range,S'_t)\leftarrow$
$active(u,r,s)\&$
$has\_relation(r',p\_range)\&$
$further\_delegatable(u',r',false)\&$
$senior(r,r'')\&$
$junior(r',r'')\&$
$can\_delegate(r'',cr,d,w)\&$
$lt(depth(u,r,valid\_TIS(u,r)),d)\&$
$lt(width(,r,valid\_TIS(u,r)),w)\&$
$in(S'_t,valid\_TIS(u,r)).$

其中 u、r 仍然分别表示转授用户、转授角色，u' 表示被授用户，r' 表示被授权限范围的源角色，p_range 是被授的部分权限范围。同规则 3-1 所表达的含义基本相似，不同之处是转授出去的是角色 r' 的一部分权限，所以 r' 与 p_range 必须要有对应关系。该规则成立必须满足的其他条件同规则 3-1。

总结及展望　时限是访问控制模型中重要的组成部分，

(下转第 282 页)

```
if (node is super-node){
  N= querypartition(O. Obj, Patition-table)
  If Plocal = N (该操作由超级节点本身负责)
       getnewObjectVersion (Oi, NCGSi-1)
else if (O 来自于自身的群)
else (该操作由团队内其它节点负责)
    Ni=SHA1(filename+ N)
    send(O, searchsite(Ni))
}else{
If Plocal = querypartition(O. Obj, Patition-table)
    getnewObjectVersion (Oi, NCGSi-1)
  else
    return;
}
```

协同结束期的工作可以由以下功能模块实现：

功能模块 6. endCollaboration：

1)某个节点结束工作后，向超级节点发送退出工作消息 send(site_exit, super-node)。

2)超级节点收到团队内所有其它节点 site_exit 消息并且本身也完工后，向其它超级节点发出团队退出协同工作消息。

3)当超级节点收到所有其它超级节点退出协同工作消息并且自己所在团队也结束所有工作时，在本团队内收集所有的文档对象，进行 merge 得到完整的文档对象 filename。

5 架构性能分析及实验

新架构原来一个站点顺序发送的操作可以同时被多个站点发送；而需要在同一站点处理的远程操作也可以分布式地发送到许多站点并发处理，类似多线程机制。由于底层机制相同，可以屏蔽底层一致性而仅仅考虑两种协同模式之间的差异。实验实现了 CHORD 环构造和查询功能，并模拟 2 种协同的交互模式。假设协同参与站点平均 5 秒产生一个本地操作，而每个站点维护远程操作处理时间为 2 秒。理论上 N 个成员的团队效率是单个站点的 N 倍，实际效率要差一点。

可以发现其它一些特性：

1)从图 4 可以看出，团队内成员数量对于团队协作是很重要的，参与成员的多少对于团队处理操作的总输出有着决定意义。

2)从图 5 可以看出，在操作基本均匀分布的情况下，通用模式的站点与团队模式的站点操作负荷差不多。新系统在不增加站点工作开销使整体工作效率得到了提高。

结论及未来工作展望 基于多专业领域的团队协同一致性模型架构是有别于通用协同模式的，它基于 Chord 和图形设计协同一致性维护算法，完善地兼顾领域内和跨领域协同，对大规模工程协同应用有其借鉴意义。

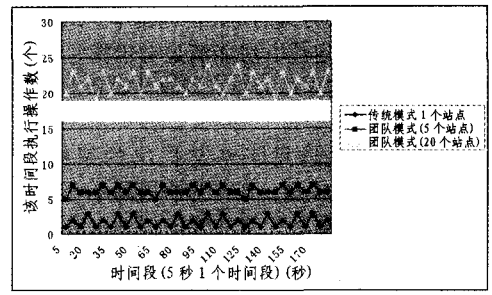

图 4 协同过程中操作输出对比(考虑通用模式，以及有 5 个和 20 个成员的团队模式)

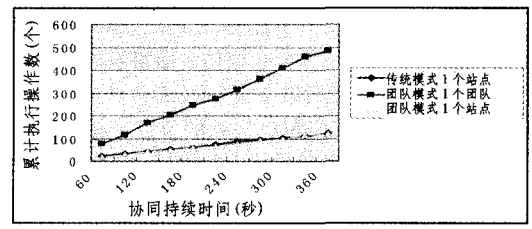

图 5 两种模式操作负荷量对比

在未来的工作中可以作如下改进：

1)可以考虑提高架构的通用性或者提供更好的封装性，使其适用于多种协同。

2)进一步研究在现有技术上如何改进从而支持群内用户的动态加入和退出。

参 考 文 献

1 Stoica I, Morris R, Karger D. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proc. Of the ACM SIGCOMM. San Diego, 2001

2 Sun C Z, Chen D. Consistency Maintenance in Real-Time Collaboration Graphics Editing Systems. ACM Transactions on Computer-Human Interaction, March 2002

3 Sun C Z, Chen D. A multi-version approach to confilct resolution in distributed groupware system. In: Proceedings of International Conference on Distributed Computing Systems

4 Gao Liping, Shao Bin, Gu Ning. Separating Data and View: Support View-wandering Between Different Trades. CSCWD'07

(上接第 279 页)

不具有时限的访问控制模型是不完备的。本文在 RBDM0 和 RDM2000 两种基于角色的委托模型基础上，提出了支持有效时间的委托授权模型，并对带时限的角色完全授权和部分授权以及增加有效时间对相应委托树结构的影响进行了讨论，并形式化地描述了他们的授权判断规则。我们还实现了基于 JAVA 的带时限的委托模型的原型，该原型充分体现了委托模型中的时效性。

在现实中对委托授权模型的时限性要求是多种多样的，进一步完善与时间相关的约束规则的定义，以及他们在委托模型中的应用是下一步研究的方向。

参 考 文 献

1 Barka E, Sandhu R. Framework for Role-based Delegation Models. In: Proceedings of 16th Annual Computer Security Application Conference, 2000. 168～176

2 Zhang Xinwen, Oh Sejong, Sandhu RS. PBDM: A Flexible Delegation Model in RBAC. In: Proceedings of the 8th ACM Symposium on Access Control Models and Technologies, 2003. 149～157

3 Sandhu R S, Coyne E J, Feinstein H L, Youman C E. Role-based Access Control Models. IEEE Computer, 1996, 29(2): 38～47

4 Barka E, Sandhu R. A Role-based Delegation Model and Some Extensions. In: the 23rd National Information Systems Security Conference, 2000. 101～114

5 Zhang Longhua, Ahn G-J, Chu Bei-tseng. A Rule-based Framework for Role-based Delegation. In: Proceedings of the 6th ACM Symposium on Access Control Models and Technologies, 2001. 153～162

6 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

7 孙波，赵庆波，孙玉芳. TRDM——具有时限的基于角色的转授权模型. 计算机研究与发展，41(7)：1104～1109

8 汤庸. 时态数据库导论. 北京：北京大学出版社

9 Zhang Xinwen, Oh Sejong, Sandhu R S. PBDM: A Flexible Delegation Model in RBAC. In: Proceedings of the 8th ACM Symposium on Access Control Models and Technologies, 2003. 149～157