第27卷第2期 华东交通大学学报 Vol.27 No.2
2010年4月 Journal of East China Jiaotong University Apr.,2010

文章编号:1005-0523(2010)02-0037-05

协作环境下支持委托的访问控制技术

向华萍[1],王英华[1],付智辉[2]

(华东交通大学 1.软件学院; 2.电气与电子工程学院,江西 南昌 330013)

摘要: 协作环境下的访问控制具有动态性和上下文相关性,传统的访问控制不能满足协作系统的特殊需求。针对协作环境的特点,从任务分解及映射、时间约束、授权委托等方面对传统的RBAC模型进行扩展,提出一个支持委托的访问控制模型(DS-RBAC)。最后给出了由多种控制机制相结合的实现方案,实现了协作组内部自主授权和动态访问控制相结合的灵活权限管理。

关 键 词: 协作环境; 授权委托; 时间约束; 角色访问控制

中图分类号: TP393.08 **文献标识码:** A

协作系统是多部门、多用户、多任务环境下的集成体系,用户为了完成共同目标互相沟通协同工作。协作系统的交互特性既要求协作用户能及时得到其所需要的资源信息,又要保证信息的安全性。因此,协作系统需要制定有效的访问控制策略,保证合法用户对授权信息进行及时访问,同时防止非法用户的侵入和合法用户对未授权资源的非法使用。

国内外很多学者都从不同角度、不同层次对访问控制进行了深入研究。Sandhu等[1]提出的基于角色的访问控制(RBAC)是一个行之有效的安全技术,可以减少访问控制的复杂性,降低系统管理的开销。文献2针对协同系统中共享资源访问控制策略的权限复杂问题,将角色访问控制策略应用到协同系统中,并设计了基于角色的系统功能权限的位映射算法来降低授权管理的复杂性。文献3提出一个专门针对协同环境下CAD模型的多层次动态的安全访问控制(MLDAC)模型,实现了权限的动态授权管理,增强了对协同设计操作的有效控制。但这些模型的访问控制策略都是由安全管理员制定,在协作系统中,为了完成共同的任务,经常需要为某些用户授予一些临时的权限,任务完成后撤销权限。如果这些临时权限的授予都要由安全管理员来实施,显然是不可行的,因此也需要让其他用户具有授权的权限。文献4支持委托授权解决了分布式系统中权限集中管理的问题,允许普通用户将一部分权限委托给他人,由其执行某些任务,任务结束后系统或委托者撤销委托的权限,但忽略了访问控制的时效问题,使得模型的描述能力受到限制。为此,本文针对上述问题对协作环境下的访问控制技术进行深入研究,提出一个支持委托的动态访问控制模型(DS-RBAC)。

1 RBAC模型

RBAC的核心思想就是将访问权限与角色相联系,通过为用户分配适合的角色,让用户与访问权限相联系。R. Sandhu等人于1996年提出了一个基于角色的访问控制参考模型,对角色访问控制产生了重要影响,被称为RBAC96模型。该模型由于系统全面地描述了RBAC多方面、多层次的意义而得到了广泛认可。RBAC96模型包括4个不同层次,分别为$RBAC_0$,$RBAC_1$,$RBAC_2$和$RBAC_3$。下面将对这个模型进行简单介绍。

$RBAC_0$包含RBAC模型的核心部分,是最基本的模型。$RBAC_0$形式化的定义如下。

定义1 $RBAC_0$模型定义了支持RBAC的最小需求,如用户、角色、权限、会话等概念。

收稿日期:2009-11-30

基金项目:江西省科技支撑计划重点项目(42108002);华东交通大学科学技术研究项目(09RJ01)

作者简介:向华萍(1976-),女,硕士,讲师,研究方向为信息安全。

●U,R,P,S 分别表示所有的用户、角色、权限以及会话的集合;

●UA$\subseteq U\times R$ 是多对多的用户角色指派关系;

●PA$\subseteq P\times R$ 是多对多的权限角色指派关系;

●user: $S\rightarrow U$ 表示映射每个会话到一个用户;

●roles: $S\rightarrow 2^R$ 表示映射每个会话 s 到一组角色 roles(s) $\subseteq\{r\mid(\text{user}(s),r)\in\text{UA}\}$,并且会话 s 拥有权限 $U_{r\in\text{roles}(s)}\{p\mid(p,r)\in\text{PA}\}$。

定义 2 RBAC$_1$ 模型包含 RBAC$_0$,并且定义了角色继承关系。RBAC$_1$ 增加了如下元素:

●RH$\subseteq R\times R$ 是 R 上的偏序关系,记为$\geqslant$,称作角色继承;

●roles: $S\rightarrow 2^R$ 修改为 roles(s) $\subseteq\{r\mid(\exists r'\geqslant r)\ [\text{user}(s),r')\in\text{UA}]\}$,同时会话 s 拥有权限 $U_{r\in\text{roles}(s)}\{p\mid(\exists r''\leqslant r)\ [p,r'')\in\text{PA}]\}$。

RBAC$_1$ 体现了 RBAC 模型中角色继承关系的语义。一个会话拥有的角色包含 UA 关系里面指定的角色以及它们的父角色,会话拥有的权限包含其拥有的所有角色在 PA 关系里面的权限以及它们的子角色对应的权限。

RBAC$_2$ 模型同样包含 RBAC$_0$,并且从语义上给出了一些约束形式的简短说明,主要包括两种约束机制: 角色互斥和角色基数约束,这也是实际系统中最通常考虑的两种限制形式。

RBAC$_3$ 包含 RBAC$_1$ 和 RBAC$_2$,自然也包含 RBAC$_0$,是一个完整的 RBAC 模型,包含一切模型元素,也是最复杂的一种模型。角色层次和约束机制同时存在,约束也可以作用在角色层次之上。

图 1 和图 2 分别给出了 RBAC96 模型层次图和基本原理图。

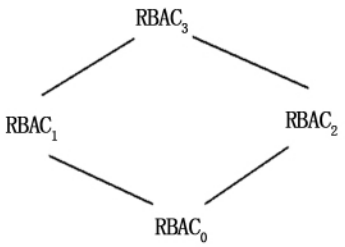

图 1 RBAC96 模型层次图

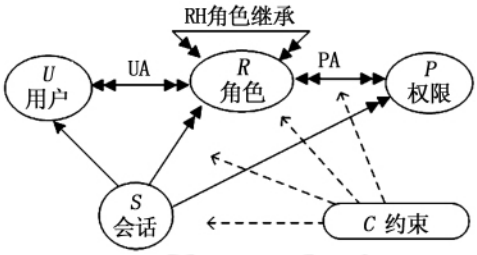

图 2 RBAC96 模型的基本原理图

2 协作环境下访问控制的特性

协作环境面向的主要对象是各部门、各企业间的协作项目,协作项目在逻辑上可分解为若干相关联的子任务,这些子任务在执行过程中需要不同的用户参与。为了更高效地执行协作项目,缩短工作时间,降低协作费用和风险,以完成协作项目中的任务为目标,安全协调各类信息资源为准则来设置虚拟组织,实现临时性动态联盟。

协作环境下虚拟组织的协同工作涉及到协作任务、协作成员、协作资源等多种因素。协作成员是协作任务的执行者; 协作任务是协作成员所要共同完成的目标; 协作资源是协作任务执行过程中涉及到的信息、工具、设备等信息。通常情况下,协作系统需要根据涉及的部门、任务的性质的特点将协作项目分解成各子任务委托给协作成员。分解规则如下: (1) 若虚拟组织中的协作成员可以独立完成某个任务,则不需要委托和分解; (2) 若虚拟组织内成员无法独立完成协作任务,可以将任务分解成更细的子任务并委托给相关协作成员共同完成; (3) 不需要再分解的任务或子任务称为基本任务,基本任务是具有最小单位的任务。协作系统中各因素的相互关系如图 3 所示。

企业在执行协同项目的过程中,协作资源处于相对开放的网络环境,非常需要有效地访问控制技术保证协作资源的安全。协作系统的访问控制应具备以下几种特性: (1) 上下文相关性。在协作任务执行过程中,协作资源的访问权限控制并不是静止不变的,而是随着执行任务的上下文环境的变化而变化,每一

步对信息的处理都与以前的处理相关,相应的访问控制也一样。(2) 时效性。在协作系统中用户通过网络共享资源,为了提高资源的使用效率,需要从时间、空间上合理安排用户的请求。另外用户拥有的权限也具有时效性。(3) 授权灵活性。协作系统的大多数任务都需要分解后再由多个成员共同完成。因此协作成员需要灵活的自主授权和撤销授权的权限。

RBAC 模型由于缺乏对时间约束、授权委托等方面的描述,不具备上述几种特性,因此本文从协作任务、时间约束、授权委托等方面对传统的 RBAC 模型进行扩展,设计并实现了支持委托的动态访问控制模型(DS – RBAC),能够很好地满足协作环境中的访问控制需求。DS – RBAC 的基本原理如图 4 所示。

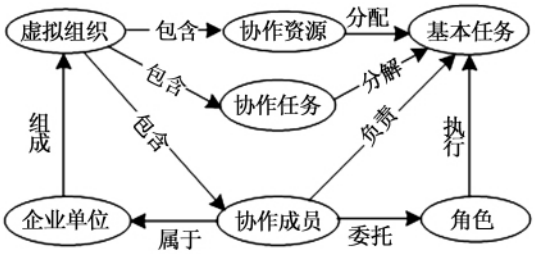

图 3　协作系统虚拟组织结构

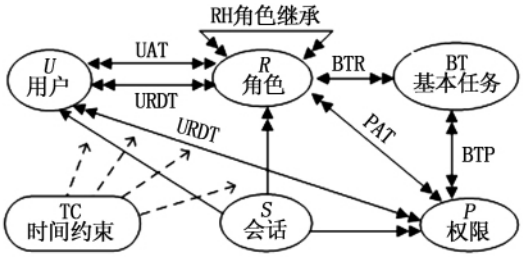

图 4　DS – RBAC 模型的基本原理

3　DS – RBAC 模型的扩充定义

3.1　动态的访问控制

虚拟组织中基本任务的访问权限不是静态的,而是依照任务执行情况以及各基本任务间的关系的变化而改变,各基本任务的执行有一定的顺序依赖关系。因此需要在协作系统中建立上下文相关信息以及基本任务、角色、权限之间的映射关系,同时引入时间约束保证协作任务的时效性。

定义 3　$BT = \{bt_1, bt_2, \cdots, bt_{n1}\}$ 表示协作任务分解后的基本任务的集合。

定义 4　$BTS \subseteq 2^{BT}$ 表示基本任务的顺序依赖关系。

定义 5　$BTR \subseteq BT \times R$ 表示基本任务与角色多对一的映射关系。基本任务是不需要再分解的任务,由一个角色独立完成,一个角色可以执行多个基本任务。

定义 6　$BTP \subseteq BT \times P$ 表示基本任务与权限多对多的映射关系。一个基本任务可以包含多个权限,一个权限也可以指派给多个基本任务。

定义 7　$T = \{t_1, t_2, \cdots, t_{n2}\}$ 表示时间的集合。用户的权限是具有时间性的,本模型通过增加时间约束,来保证用户操作的时效性。

定义 8　$UAT \subseteq U \times R \times T$,表示多对多的具有时间约束的用户角色指派。

定义 9　$PAT \subseteq P \times R \times T$,表示多对多的具有时间约束的角色权限指派。

3.2　授权委托

授权委托机制能够为协作成员提供灵活的自主授权。协作任务完成后,系统将撤销委托者委托的权限,以满足安全访问控制中“最小特权原则”。

定义 10　$URDT \subseteq U \times U \times R \times T$ 是多对多的具有时间约束的角色委托。角色委托 $(u_1, u_2, r, t) \in URDT$,是一个四元组,表示委托者 u_1 在时间 t 内将角色 r 委托给受托者 u_2。

定义 11　$UPDT \subseteq U \times U \times P \times T$ 是多对多的具有时间约束的权限委托。权限委托 $(u_1, u_2, p, t) \in UPDT$,是一个四元组,表示委托者 u_1 在时间 t 内将权限 p 委托给受托者 u_2。

用户可以通过用户角色指派关系 UAT 和角色权限指派关系 PAT 由安全管理员为用户分配角色,从而获得相应权限; 也可以通过委托机制 URDT 和 UPDT 由协作项目负责人临时委派角色或权限来完成指定任务。

4 DS – RBAC 的实现

DS – RBAC 模型能够较好地解决协作环境下用户访问控制问题，提高系统的灵活性和权限管理效率。DS – RBAC 的实现是由多种机制协作完成，包括时间约束机制、安全策略管理机制、授权委托机制、上下文约束机制、授权验证机制等。

4.1 时间约束机制

时间约束包括周期时间约束 CT 和区间时间约束 RT。CT 可以由一个四元组(m,dM,dW,h)实现，m表示月份约束，是一个具有 12 位有效字节的二进制数，可以转换成整数，例如 5 的二进制 101，表示有效月份是 1 月和 3 月；dM 和 dW 都是日约束，dM 用来指定每月的哪几天，dW 表示每个星期的哪几天，dM 和 dW 是互斥的，只需要设置其中一个，不需设置的元素用“\”表示；h 是小时约束，指定每天可以执行操作的时间。dM，dW 和 h 都是用二进制表示，也可以转换成整型。(m,dM,dW,h) 的四个元素的值也可以是通配符“* ”，表示所有的意思。

RT 是一个二元组(t_{start},t_{end})用来指定整个有效时间的起始和结束时间，t_{start}和 t_{end}是三元(y,m,d)指明年月日。一个时间约束同时需要指明周期时间约束和区间时间约束。

例如，$rt_1=((2008,3,1),(2010,12,31))$，$ct_1=(60,\backslash,31,*)$(即 $ct_1=(111100,\backslash,11111,*)$)，则时间 $t_1=rt_1\cap ct_1$ 表示从 2008 年 3 月 1 号到 2010 年 12 月 31 号的 3 至 6 月的(工作日)。

4.2 安全策略管理机制

安全策略管理机制是访问控制模型的基础部分，为安全管理员提供一个定义和维护系统安全策略的接口，主要包括对角色和权限的管理与维护，设置用户角色指派和角色权限指派。在实际中，用户的操作都具有时效性，因此管理员在设置的用户角色指派和角色权限指派是受时间约束的。例如，如果 Mike 要在上述的 t_1 时间担任财务处的出纳职务，则安全管理员需要将出纳的角色 r_1 指派给用户 Mike，即有：$(Mike,r_1,t_1)\in UAT$。

4.3 上下文约束机制

上下文约束机制主要是解决协作环境下多个用户共同完成协作项目的问题。上下文约束机制主要的工作包括将协作项目分解成若干基本任务，设置基本任务角色映射关系和基本任务权限映射关系，以及设置基本任务的顺序依赖关系。上下文约束机制的实施是由项目管理员负责。例如，根据需要将协作项目分解为 5 个基本任务 $BT=\{bt_1,bt_2,\cdots,bt_5\}$，角色 r_2 可以执行 bt_1 和 bt_2 两个任务，则有$(r_2,bt_1)\in BTR$，$(r_2,bt_2)\in BTR$。若设置$\{bt_1,bt_3,bt_4\}\in BTS$，表示这三个任务的执行顺序是：$bt_1\rightarrow bt_3\rightarrow bt_4$。

4.4 授权委托机制

项目管理员通过上下文约束机制分解协作项目并作相应的设置后，系统将会增加一些新的临时的角色和权限，如果这些角色和权限的指派都由安全管理员执行，必将加重安全管理员的负担，而且安全管理员对具体项目的了解程度是不及项目管理员。授权委托机制就是为项目管理员或其他用户提供一个委托角色或权限的机制。如果具有委托权的用户 Jim 设置$(Jim,Alice,r_2,t_2)\in URDT$(其中 t_2满足上述时间约束的设置)，表示用户 Jim 将角色 r_2委托给用户 Alice，则 Alice 在时间 t_2内就拥有 r_2包含的权限。

4.5 授权验证机制

用户登录后与系统建立会话，系统先获取该用户所有的权限，组成一个权限时间约束集，此权限时间约束集给出该用户被授权的所有权限及相应的约束时间，权限时间约束集中的权限并非都是有效权限，有效与否取决于当前时间是否在权限对应的有效时间内，以及权限所属的基本任务的顺序依赖关系。

授权验证机制包括两个过程：用户权限获取过程和权限有效性验证过程。(1) 用户权限获取过程：递归遍历用户的所有角色及其子角色，以及角色包含的基本任务，遍历到权限时，计算权限的有效时间，然后把权限及有效时间加入到权限时间约束集中。(2) 权限有效性验证过程：递归遍历用户下的权限时间约束集，找到相应权限，并判断其有效时间是否包含当前时间，满足要求返回权限有效，否则返回权限无效。

5　实例分析

结合某高校综合信息系统中对该模型的应用进行简要说明。高校综合信息系统是一个对高校所有资源信息进行统筹管理,为学校各部门之间信息交流和协同工作提供数据支持,由若干个子系统集成的综合信息系统。系统中很多任务不能由一个部门独立完成,必须由多个部门协同合作。例如,教务处根据市场调查,结合软件专业与应用背景专业接轨的思想,拟新增加一个"软件 + 桥梁"专业,其相关工作必须由教务处、软件学院和土木学院共同完成,因此三个单位相关人员需要临时组成一个相互协作的虚拟组织。首先由教务处(即项目管理员)分解任务,并完成各种指派和映射关系,虚拟组织的其他人员得到授权后在规定时间内完成自己的子任务。高校综合信息系统运行结果表明,DS - RBAC 模型能够有效解决多部门、多用户的集成系统中跨部门的协作项目的访问控制问题,实现协作环境下的临时性动态联盟和安全资源共享。

6　结束语

随着计算机网络的飞速发展,多领域大范围的协同合作成为一种趋势,不同组织为了共同的利益建立协作关系,研究多领域协作环境下的访问控制模型成为必要。本文设计的支持委托的角色访问控制模型能够满足多领域协作环境下访问控制的上下文相关性、时效性以及授权灵活性等特殊需求,具有丰富的访问控制的描述能力,有效地解决了协作环境中不同部门、不同企业间的相互协作以及资源互访问题。

参考文献:

[1] SANDHU R S,COYNEK E J,FEINSTEINK H L,et al. Role - based access control Models [J]. IEEE Computer,1996,29(2):38 - 47.

[2] 付喜梅. 协同环境中基于 RBAC 模型的访问控制策略 [J]. 计算机工程,2009,35(11):140 - 142.

[3] 方萃浩,叶修梓,彭维,张引. 协同环境下 CAD 模型的多层次动态安全访问控制 [J]. 软件学报,2007,18(9):2 295 - 2 305.

[4] 魏永合,王成恩,马明旭. 工作流系统中的委托授权机制研究 [J],计算机集成制造系统,2009,15(1):160 - 165.

[5] 李黎青. 财政收费系统中的信息安全策略研究 [J]. 华东交通大学学报,2005,22(5):121 - 123.

Delegation - supported and Role - based Access Control in Collaborative Environment

Xiang Huaping[1],Fu Zhihui[1],Wang Yinghua[2]

(1. School of Software; 2. School of Electrical and Electronic Engineering,East China Jiaotong University,Nanchang 330013,China)

Abstract: Access control in collaborative environment has dynamic characteristic and contextual relevance. However,traditional role - based access control is far from satisfying demands of the collaborative system. Aiming at the characteristics of collaborative circumstance,a delegation - supported and role - based access control(DS - RBAC) is proposed which extends traditional RBAC from task decomposing and mapping,time constraints and delegation. Finally,an implementation scheme,which combines a variety of control mechanisms,is designed to achieve flexible rights management with internal self - authorization for collaborative groups and dynamic access control.

Key words: collaborative environment; delegation; time constraint; RBAC

(责任编辑　刘棉玲)