第 29 卷 第 7 期
2006 年 7 月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 29 No. 7
July 2006

一种基于角色代理的服务网格虚拟组织访问控制模型

孙为群[1),2)] 单保华[1),2)] 张 程[1),2)] 刘 晨[1),2)]

[1)](中国科学院计算技术研究所网格与服务计算中心 北京 100080)

[2)](中国科学院研究生院 北京 100039)

摘 要 给出一种基于角色代理技术的虚拟组织访问控制模型,与同类研究成果相比,在不降低自治域的安全管理效率的情况下,能够实现虚拟组织的细粒度授权和确保自治域的安全策略不被破坏.该模型的一个原型系统已经实现,并通过一个基于网格的低成本电子政务平台中的实例进行了验证.

关键词 服务网格;虚拟组织;访问控制;角色代理;策略

中图法分类号 TP311

A Role-Based Delegation Access Control Model for Virtual Organization in Service Grid

SUN Wei-Qun[1),2)] SHAN Bao-Hua[1),2)] ZHANG Cheng[1),2)] LIU Chen[1)]

[1)](*Research Center for Grid and Service Computing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080)

[2)](*Graduate University of Chinese Academy of Sciences, Beijing* 100089)

Abstract In a service grid, the efficiency of security administration and security assurance of virtual organizations and autonomous domains are challenging issues. Access control is usually implemented through mapping virtual organization users to autonomous domain's local users, which reduces the efficiency of security administration and bears the risk of violating security policy of these domains applying RBAC model. A role-based delegation access control model for virtual organizations is proposed in the paper, which has little effect on security administration efficiency, and does not violate autonomous domain's security policy. It is implemented and validated in a grid-based e-Government platform.

Keywords service grid; virtual organization; access control; role-based delegation; policy

1 引 言

网格计算是研究在动态的、多自治域的虚拟组织中如何高效地共享资源和协同求解的问题.虚拟组织(Virtual Organization,VO)是网格计算的关键概念,是由个人和自治域按照一定的资源共享规则形成的集合.自治域或个人通过虚拟组织共享资源和进行问题求解[1].

以面向服务的体系结构(Service Oriented Architecture,SOA)和 Web 服务为代表的面向服务的计算,是目前分布式计算技术最新的方向.SOA 是一种基于服务组织计算资源的软件体系结构风格.服务则是一种自治、开放、自描述和平台独立的

收稿日期:2006-02-17;修改稿收到日期:2006-05-17.本课题得到国家自然科学基金(90412005)资助.孙为群,男,1975 年生,博士研究生,主要研究方向为服务网格安全技术.单保华,男,1978 年生,博士研究生,主要研究方向为面向服务的软件集成、业务端编程技术和服务虚拟组织建模.张 程,男,1977 年生,博士研究生,主要研究方向为动态服务组合、语境敏感计算技术.刘 晨,男,1980 年生,博士研究生,主要研究方向为服务网格与软件集成、服务语义.

网络构件[2].

服务网格是网格计算技术和面向服务的架构相互融合的结果,是把服务作为资源来构建网格,以支持服务的共享、管理和有效访问. IBM 从商业服务的角度出发,提出服务域(service domain)技术来构建商业服务网格解决方案①. 中国科学院计算技术研究所的中德联合软件实验室从最终用户编程的角度出发,以业务服务为核心来构建业务服务网格解决方案[3]. 服务网格的虚拟组织既有动态性——不同自治域的服务和用户随时随地地加入和退出,又有自治性——服务提供者自主且可动态地决定哪些用户可以在什么样的条件下来访问共享的服务.

传统的访问控制模型难以很好地解决虚拟组织的动态性和自治域的自主性带来的安全问题. 本文提出一种基于角色代理的虚拟组织访问控制模型(Virtual Organization Role-based Delegation, VORD)来解决此问题. 本文第 2 节介绍角色代理的相关研究工作和本文的动机;第 3 节以形式化方式详细介绍本模型的组件和操作,并给出模型的原型实现和一个应用实例;第 4 节是总结本模型的特点和下一步的工作.

2 研究动机和相关工作

2.1 问题的提出

目前参与虚拟组织进行协作和资源共享的自治域是采用自主访问控制. 虚拟组织访问控制的方法是以 Globus 为代表的用户标识(identity)映射方法[4,5],即自治域共享资源的同时,为虚拟组织建立一个有权使用共享资源的本地用户 U_{VO}. 虚拟组织的任何用户访问该资源的时候,都将以 U_{VO} 账户来访问. 如果这种方法应用到由采用 RBAC 自治域组成的 VO,会加重自治域的安全管理工作,并带来安全问题.

比如,自治域有 N 个资源要共享给 VO,并且本地安全策略要求一个用户只能拥有访问一个资源的权限,那么自治域不得不在本地建立 N 个用户, $U_{VO_1}, U_{VO_2}, \cdots, U_{VO_N}$,并将这 N 个用户映射到虚拟组织. 同时在和 VO 协定时,需要声明 VO 只能授权一个用户拥有访问一个资源的权限.

上述过程中存在两个缺点. 首先,自治域可能随时向 VO 共享或撤销有类似安全需求的资源. 管理员要随时创建/删除用户和映射关系,管理工作繁琐. 其次,如果 VO 安全管理员错误地把访问两个资源同时分配给一个用户,由于自治域的访问控制机制是通过角色来判断用户有无访问权限的,无法检查出这种错误,从而造成自治域的安全漏洞.

出现上述问题的根本原因是,虚拟组织访问控制是基于标识的映射方法,而自治域的访问控制是基于构成访问控制策略基础的语义构造子——角色[6]. 映射方式屏蔽掉了自治域访问控制核心组件——角色,也就屏蔽掉了自治域的访问控制策略,从而带来授权工作繁重和自治域安全漏洞等问题.

本世纪初,被广泛研究和应用的代理和 RBAC 技术相结合形成了角色代理技术. 这种技术既具有 RBAC 的授权管理高效、策略表达灵活且易于实现等特点[6,7],又具有代理技术的安全管理安全地分散化,即将部分授权工作下放给单独的用户,系统通过预先定义的代理策略来防止单独用户滥用授权能力. 角色代理技术为解决类似上述的虚拟组织访问控制问题提供了新的思路和技术基础.

2.2 基于角色的代理技术

Barka 和 Sandhu 首先提出用户到用户的角色代理模型 RBDM0[8]. 该模型是在 RBAC96 模型和 ARBAC97 模型基础之上进行扩展得来的. 作者采用系统方法,通过 8 个标识性特性(identified characteristics)来描述 RBDM 模型的框架,并分析了这 8 个特性组合中有意义的组合. RBDM0 没有对角色继承、角色权限的部分代理和多级代理(multi-step delegation)进行探讨.

基于规则的角色代理模型 RDM2000 是 RBDM0 的扩展,通过引入代理关系来支持角色继承和多级代理,并且提出一种基于规则的策略描述语言来描述和实施安全策略[9]. 文献[9]给出了 RDM2000 的形式化描述,并用代理树直观地描述角色的代理过程. RDM2000 不支持角色权限的部分代理,使得该模型的权限控制粒度较粗.

Zhang 等对 RDM2000 进行扩展,提出的 PBDM 模型族支持细粒度的权限代理、更高的代理可控性和更加高效的代理过程[10]. PBDM0 模型支持角色权限的部分代理是通过定义临时代理角色 R_{temp},并将欲代理的角色的部分权限分配给临时代理角色 R_{temp},然后将 R_{temp} 代理给相应的用户. PBDM1 模型通过将代理角色分成初始角色、可代理角色和代理

① http://www-128.ibm.com/developworks/cn/grid/gr-servicegrid.

角色来支持在安全管理员监控之下的用户代理. PBDM2 支持角色到角色的代理，使得系统可以一次把某个角色代理给多个用户，提高了代理的效率. PBDM 没有对临时代理角色和代理角色以及临时代理角色之间的关系进行探讨，其复用性以及可管理性不强，而且 PBDM 模型族都是集中安全控制的模型[11].

赵庆松等对 RDM2000 模型进行扩展，提出重复和部分角色代理模型 RPRDM[12]. 作者通过对每一个代理角色赋予一个屏蔽值(mask)来实现部分代理. 这种方式虽然理解起来直观，但没有对部分代理角色之间的关系以及部分代理角色和代理角色之间的关系进行深入的探讨，部分代理角色的可复用性和可控性都不强.

Lee 等提出一种 sub-role 层次组件来实现受限的代理[13]. 但是 sub-roles 的分类固定，模型只能实现粒度较粗的部分权限代理，而且 sub-roles 之间的关系也是固定，大大降低了模型的灵活性.

虚拟组织用户对资源的访问，需要经过虚拟组织和共享该资源的自治域两个管理域的控制. 而目前的角色代理模型主要集中在一个管理域的角色代理研究，不适用于虚拟组织的访问控制，研究跨自治域的角色代理模型是非常必要的.

3　基于角色代理的虚拟组织访问控制模型

3.1　模型概述

模型的基本思想是，当参与虚拟组织 VO 的自治域(Autonomous Domain，AD)把资源共享给 VO 的时候，同时将有权访问这个资源的角色 R_{AD}(一个或多个)代理给 VO. 如果 VO 中的用户 U_{VO} 欲访问 AD 提供的资源，可以向 VO 提出签发证书请求. 如果根据 VO 安全策略允许用户 U_{VO} 访问该资源，则 VO 将角色 R_{AD} 作为代理内容签发一个角色代理证书给 U_{VO}. 用户 U_{VO} 每次访问虚拟资源的时候，出示 VO 签发给他的角色代理证书. AD 首先将请求转发给 VO，VO 根据自己的安全策略对请求判定，并将判定结果返回给自治域；如果判定结果为允许，自治域再根据自己的访问控制策略对请求进行判定，如果合法，则允许访问；否则拒绝. 模型基本思想如图 1 所示，代理和访问过程用序号标出.

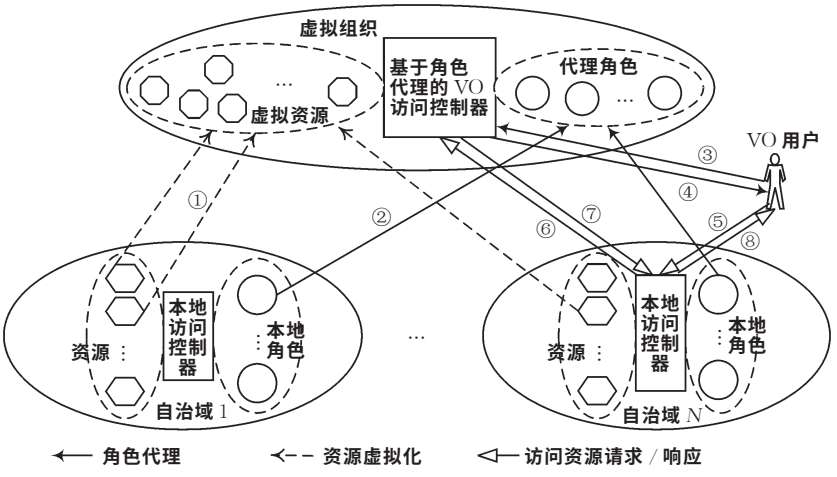

图 1　模型原理示意图

在上述过程中，用户的访问控制经过虚拟组织和自治域两层访问控制，而且在访问自治域资源的时候，出示的证书是以自治域角色为内容的，自治域的安全策略可以据此判定其合法性，保证了自治域的安全不会受到破坏.

从图 1 可以看出，在虚拟组织中存在许多自治域代理给虚拟组织的角色，我们称其为代理角色. 虚拟组织需要通过管理和代理这些角色给虚拟组织的用户，来实现对所有共享的资源进行授权和访问控制. 本文给出一个模型，可以很好地管理这些角色的代理、分解和撤销等，实现动态开放的虚拟组织访问控制.

3.2　模型组件

RBAC96 模型是一种被信息安全领域广泛认可的访问控制模型. RDM2000 有着良好的形式化描述，并且作者给出了基于规则的授权策略规范语言，使得模型的可操作性很强，易于实现. 本模型在 RBAC96 和 RDM2000 的基础上进行扩展，提出了分解角色和虚拟组织角色. 分解角色用来解决虚拟组织对角色部分授权的细粒度访问控制需求. 虚拟

组织角色主要用来表达虚拟组织访问控制策略的约束条件.

模型如图 2 所示. 共享资源给虚拟组织的自治域将角色代理给虚拟组织,组成了虚拟组织的代理角色. 由于同一自治域内的不同代理发起者将不同的角色代理给虚拟组织,这些角色之间可能存在着层次(继承)关系.

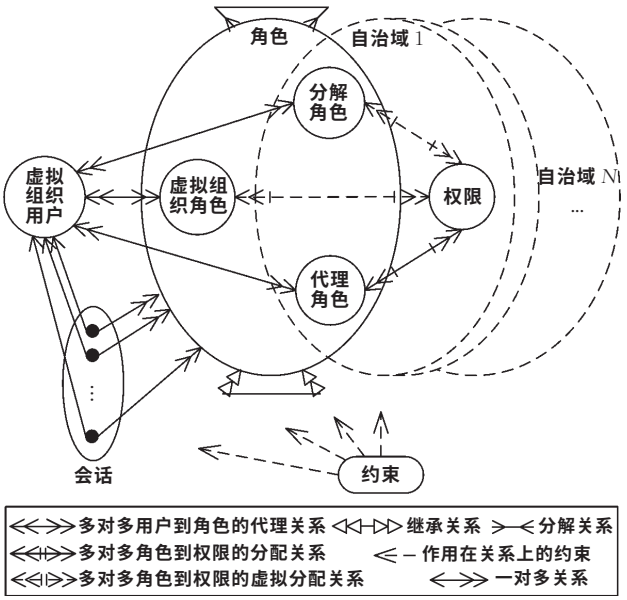

图 2 VORDM 模型

虚拟组织管理员或有分解权限的用户为了对代理角色进行部分代理,在虚拟组织范围内对代理角色进行分解而形成的角色叫分解角色,和代理角色存在着多对多的分解关系.

虚拟组织角色继承代理角色或分解角色,实现大粒度的权限聚集.

下面给出 VORDM 的形式化描述. 定义 1 给出了角色分解的定义;定义 2 是对 RDM2000 模型进行扩展得到的基本组件;定义 3 是三个常用的约束的形式化描述.

其中,P,R,DR,DDR,VR,U,S 分别代表以下集合:权限(Permissions)、角色(Roles)、代理角色(Delegation Roles)、分解代理角色(Decomposition Delegation Roles)、虚拟组织角色(VO Roles)、虚拟组织用户(VO Users)和会话(Sessions).

定义 1. 对于 $P(r)\neq\varnothing$ 的角色 r,若存在一个角色集合 $A=\{r_1,r_2,\cdots,r_n\}$ 满足:

(1) $A\neq\varnothing$;

(2) $P(r_j)\neq\varnothing$,且 $P(r_j)\subset P(r)$, $j=1,2,\cdots,n$;

(3) $\forall r_i,r_j\in A$, $i,j=1,2,\cdots,n$,并且 $i\neq j$,则 $P(r_i)\cap P(r_j)=\varnothing$;

(4) $\bigcup P(r_i)=P(r)$, $i=1,2,\cdots,n$,

则称集合 A 是角色 r 的一个分解. A 中的每一个元素称为分解因子,也称为 r 的分解角色. 将 r 和其分解因子 r_i 间的关系称为分解关系,记作 $r_i\lhd r$,或 $r\rhd r_i$.

推论 1. 如果 r_d 是角色 r 的一个分解因子,r_{dd} 是 r_d 的一个分解因子,则 r_{dd} 也是 r 的一个分解因子.

定义 2. 基本组件.

(1) 角色分配关系及相关函数

$UA\subseteq U\times R$ 是多对多的用户到角色的分配关系;

$UAD\subseteq U\times DR$ 是多对多的用户到代理角色的分配关系;

$UAV\subseteq U\times VR$ 是多对多的用户到虚拟组织角色的分配关系;

$UADD\subseteq U\times DDR$ 是多对多的用户到分解代理角色的分配关系;

$UA=UAD\cup UAV\cup UADD$;

$User:R\rightarrow 2^U$ 是每个角色映射到一个用户集合的函数;

$Users_D(r)=\{u\mid(u,r)\in UAD\}$,表示代理角色的用户成员集合;

$Users_V(r)=\{u\mid(u,r)\in UAV\}$,表示虚拟组织角色的用户成员集合;

$Users_DD(r)=\{u\mid(u,r)\in UADD\}$,表示分解代理角色的用户成员集合;

$Roles:U\rightarrow 2^R$ 是将用户集合 U 映射到角色集合 R 的函数,$Roles(u_i)=\{r\in R\mid(u_i,r)\in UA\}$;存在角色继承关系时,则为 $Roles^*:U\cup P\cup S\rightarrow 2^R$, $Roles^*(u_i)=\{r\in R\mid(\exists r'\geqslant r)[(u_i,r')\in UA]\}$.

(2) 权限分配关系

$PA\subseteq P\times R$ 是多对多的权限到角色的分配关系;

$PAD\subseteq P\times DR$ 是多对多的权限到代理角色的分配关系;

$PAV\subseteq P\times AV$ 是多对多的权限到虚拟组织角色的分配关系;

$PADD\subseteq P\times DDR$ 是多对多的权限到分解代理角色的分配关系;

$PA=PAD\cup PAV\cup PADD$;

$Roles:P\rightarrow 2^R$ 是将权限集合 P 映射到角色集合 R 的函数,$Roles(p_i)=\{r\in R\mid(p_i,r)\in PA\}$;存在角色继承关系时,则 $Roles^*:P\cup S\rightarrow 2^R$, $Roles^*(p_i)=\{r\in R\mid(\exists r'\leqslant r)[(p_i,r')\in PA]\}$.

(3) 角色间的关系

$R=DR\cup DDR\cup VR$；

$RH\subseteq R\times R$ 是角色 R 上的偏序关系(也称继承关系)，$r'\geqslant r$ 或 $r\leqslant r'$ 表示 r' 继承 r；

$DRH\subseteq DR\times DR$ 是 DR 上的偏序关系；

$VDRH\subseteq VR\times DR$ 是 VR 到 DR 上的偏序关系；

$VDDRH\subseteq VR\times DDR$ 是 VR 到 DDR 上的偏序关系；

$RD\subseteq DR\times DRR$ 是多对多的代理角色到分解代理角色间的分解关系；

$Decomposes: DR\rightarrow 2^{DDR}$ 是将每个代理角色映射到一个分解角色集合关系的函数；

$RR\subseteq R\times R$ 是多对多的角色到角色间的等级关系. 如果 $P(r_2)\subset P(r_1)$，则 $(r_1, r_2)\in RR$，称 r_1 高于 r_2，记作 $r_1\ltimes r_2$ 或 $r_2\rtimes r_1$. r_1 称做 r_2 的高级角色，r_2 称做 r_1 的低级角色.

推论 2.

如果 $r_1\geqslant r_2$，则 $r_1\ltimes r_2$；

如果 r_2 是 r_1 的一个分解因子，则 $r_1\ltimes r_2$；

如果 $r_1\geqslant r_2$，r_3 是 r_2 的一个分解因子，则 $r_1\ltimes r_3$；

如果 $r_1\geqslant r_2$，$\{dr_{11}, dr_{12}, \cdots, r_2, dr_{1m}\}$ 是 r_1 的一个分解，$\{dr_{21}, dr_{22}, \cdots, dr_{2n}\}$ 是 r_2 的一个分解，则 $\{dr_{11}, dr_{12}, \cdots, dr_{21}, dr_{22}, \cdots, dr_{2n}, dr_{1m}\}$ 也是 r_1 的一个分解.

(4) 动态关系

$User: S\rightarrow U$ 是将每个会话映射到一个用户的函数；

$Sessions: U\rightarrow S$ 是将每个用户映射到一个会话集合的函数；

$Permissions: S\rightarrow 2^P$ 是从 PA 推导出的一个函数，将每个会话 s_i 映射到一个权限集合

$$Permissions(s_i)=\{p\mid(\exists r'\geqslant r)(p, r')\in PA, r'\in Role(s_i)\};$$

$Roles: S\rightarrow 2^R$ 是将会话集合 S 映射到角色集合 R 的函数，$Roles(s_i)=\{r\in R\mid(User(s_i), r)\in UA\}$；存在角色继承关系时，$Roles^*: U\cup P\cup S\rightarrow 2^R$，$Roles^*(s_i)=\{r\in R\mid(\exists r'\geqslant r)[(u_i, r')\in UA]\}$.

(5) 代理关系

$DLGT\subseteq UA\times UA$ 是多对一的代理关系；

$DT\subseteq UA\times UA$ 是代理树(Delegation Tree)；

$Prior: U\cup R\rightarrow U\times R$ 是将每个 UA 映射到其它 UA 或空集的函数，$Prior(u, r)=\{(u', r')\mid(u, r)\in UA, (u', r', u, r)\in DLGT\}$.

代理路径是用户角色分配关系的有序集合，形式化描述为

$$Path(u_0, r_0)=\{(u_0, r_0), (u_1, r_1), \cdots, (u_i, r_i), \cdots, (u_n, r_n)\mid(u_i, r_i)=Prior(u_{i-1}, r_{i-1})\in UA, i>0\}.$$

定义 3.　基本约束(constraints)是用来描述授权策略规则的基本元素. X 为集合，$ANYONE(X)=x_i, x_i\in X$，表示 X 集合中的任意一个元素.

(1) 互斥(Mutex)约束

同一个用户最多被分配属于该集合中的 n 个角色，n 为大于 0 的自然数. 这样的集合也称作互斥角色集合.

形式化描述：$2^R\rightarrow N$ 是互斥角色集合映射到一个自然数的函数，即 $MR=\{mr_1, mr_2, \cdots, mr_n\}$，$mr_i=\{r_1, r_2, \cdots, r_t\}\subseteq R$，$|Roles(ANYONE(U))\cap Roles(ANYONE(MR))|\leq N$.

严格互斥假设：如果代理角色 dr_i 是互斥集合 $A=\{dr_1, dr_2, \cdots, dr_i, dr_{i+1}, dr_{i+2}, \cdots, dr_n\}$ 中的元素，且 $\{dr_{i1}, dr_{i2}, \cdots, dr_{im}\}$ 是 dr_i 的一个分解，则 $\{dr_1, dr_2, \cdots, dr_{ij}, dr_{i+1}, dr_{i+2}, \cdots, dr_n\}$ $(j=1, 2, \cdots, m)$ 也是互斥集合. 这条假设是为了确保自治域的和互斥集合 A 相关的安全策略不会被分解关系所破坏.

(2) 势(Cardinality)约束

代理角色的势是指一个角色最多可以被代理多少次.

形式化描述：$Cardinality: C(dr)\rightarrow N$ 是将代理角色的势映射到自然数的函数，即

$$|Roles^*(ANYONE(U))|\leq N.$$

推论 3.　如果 $\{ddr_1, ddr_2, \cdots, ddr_n\}$ 是代理角色 dr 的一个分解，则

$$Cardinality(dr)=\sum Cardinality(ddr_i).$$

(3) 先决(Prerequisite)约束

如果用户可以被代理角色 R_A，当且仅当用户是角色 R_B 的成员，称 R_B 是 R_A 的先决角色(Prerequisite Role)，记作称 $R_B(PR)R_A$.

形式化描述：$PR\subseteq R\times R$ 是集合 R 上的先决关系，如果 $PR=\{pr_1, pr_2, \cdots, pr_s\}$，$pr_i=\langle r_{i1}, r_{i2}\rangle$，$r_{i1}, r_{i2}\subseteq R$，$|Roles(ANYONE(PR))\cap Roles(ANYONE(U))|=0$ 或 2. 从定义可知 PR 满足反自反性、反对称性和传递性.

推论 4.　如果 $dr_1, dr_2\in DR$，且 $dr_2(PR)dr_1$，$\{ddr_{11}, ddr_{12}, \cdots, ddr_{1n}\}$ 是 dr_1 的一个分解，则 $dr_2(PR)ddr_{1i}$，$i=1, 2, \cdots, n$.

3.3　分解和代理授权操作

虚拟组织的安全管理员通过设置分解和代理授

权策略规则,来防止虚拟组织用户滥用其分解和代理权限,以使虚拟组织的安全策略能够被强制执行.在给出这些授权的形式化描述之前需要先给出一个重要的概念——先决条件(prerequisite condition).

先决条件是 ARBAC97 的关键概念之一[15].在角色代理模型中,先决条件是描述谁有权代理角色的基本概念,所以本模型仍采用 ARBAC97 模型中的描述方式.

定义 4. 先决条件是用连接符 &(与)和|(或)连接 cr 项的布尔表达式,cr 可以是 x(成员)或者 $-x$(非成员),x 是角色.

3.3.1 角色分解

角色分解是为了解决虚拟组织环境下,角色代理的部分授权问题,也就是对代理角色权限的更细的抽象划分,以解决细粒度的授权和访问控制.当虚拟组织管理员或者代理角色的成员只想将角色 r 的部分权限 $p \subset Permission(r)$ 代理给用户时,可以将代理角色分解成 dr_1 和 dr_2,并将 p 分配给分解代理角色 dr_1,$Permission(dr)-p$ 分配给 dr_2.分解角色只在虚拟组织范围内可见,自治域并不知道分解角色的存在.

下面的关系描述了对代理角色进行分解的授权.

定义 5. $can_decompose \subseteq R \times CR$,$R$ 表示代理角色或者分解代理角色集合,CR 表示先决条件集合.

$(r, cr) \in can_decompose$ 表示如果用户 u 是角色 r 的成员,并且满足 cr 条件,则可以分解角色 r.$(r, \varnothing)$表示只要是 r 的成员就可以对 r 进行分解.假如 $r_{\mathrm{VO_admin}}$是虚拟组织管理员角色,$(r, r_{\mathrm{VO_admin}})$表示只有 VO 管理员角色的成员可以分解角色 r.

3.3.2 角色代理

资源提供者将物理资源注册成虚拟资源后,必须将有权使用该资源的本地角色代理给虚拟组织,这样才能实现资源共享.虚拟组织就可以按照自己的策略将这个角色代理给组织内的用户,资源就可以被访问.代理关系表示如下.

定义 6.

$$can_delegate \subseteq R \times CR \times C \tag{1}$$

或

$$can_delegate \subseteq R \times CR \times M \times N \tag{2}$$

其中 R,CR,M,N 和 C 分别是角色(代理角色和分解代理角色)集合、先决条件集合、代理宽度的最大值、代理深度的最大值和代理势的最大值.$(r, cr, c) \in can_delegate$ 表示:如果用户 u_1 是角色 r(或 r 的高级角色)的成员,并且 u_2 的角色成员关系满足先决条件 cr,则用户 u_1 可以将角色 r 代理给用户 u_2,且代理的势不大于 c.$(r, cr, m, n) \in can_delegate$ 表示:如果用户 u_1 是角色 r(或 r 的高级角色)的成员,并且 u_2 的角色成员关系满足先决条件 cr(由虚拟组织角色组成的逻辑表达式),则用户 u_1 可以将角色 r 代理给用户 u_2,且代理的宽度和深度分别不大于 m 和 n.

式(1)主要用于资源提供者代理角色给虚拟组织,因为资源提供者通常不关心角色代理给 VO 后,形成的代理树的深度和宽度.但会从负载角度考虑代理角色势的大小,即允许 N 个虚拟组织用户同时访问共享给该虚拟组织的资源.式(2)主要用于虚拟组织内部粒度更细的代理.

资源提供者 RP 和虚拟组织 VO 协商好共享哪些资源后,把有权使用这些资源的角色 R 代理给 VO 安全管理员角色 $Administrator$,形成代理关系 $((RP, R), (VO, Administrator))$.当虚拟组织决定授予虚拟组织中的用户 U 访问这些资源时,只要将资源提供者代理的角色再次代理给该用户即可,形成代理关系如——$((Mike, R), (U, R_1))$,$Mike$ 是角色 $Administrator$ 成员.

由于虚拟组织具有跨组织特点,如果资源提供者将相应的角色直接代理给虚拟组织中的用户,虚拟组织将无法掌握这种代理是否符合虚拟组织的安全策略,而且也无法对虚拟组织资源的访问情况做出正确的审计追踪.所以资源提供者必须先将角色直接代理给虚拟组织,而不能直接代理角色给虚拟组织用户.

图 3 是代理角色 R 的一个代理树实例.自治域 A 中的用户 U_{AD} 将角色 R 代理给虚拟组织 VO_1 角色 R_{Admin} 的成员 U_1,用户 U_1 根据虚拟组织 VO_1 的代理策略,将角色代理给其它角色的成员用户,一级级地代理后形成图 3 中的代理树.从图 3 中可见,所有角

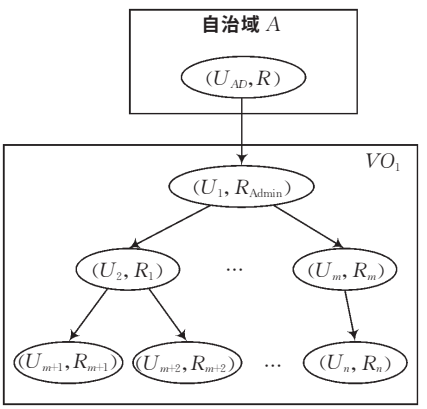

图 3 角色代理树实例

色的代理树的根结点只有一个子结点.

3.3.3　撤销角色代理

撤销是将代理给用户的角色撤销，以便用户不再拥有这个角色所赋予的权限. 角色的撤销有两种方式：时限撤销和用户主动撤销.

时限撤销方式是将每个角色代理关系和一个有效期相关联，一旦超过这个有效期，系统将自动撤销角色的代理. 本模型对时限撤销方式的定义借用具有时限的角色代理模型 TRDM[15] 中的定义.

定义 7. $can_auto_revoke \subseteq R \times Time$，$(r,t) \in can_auto_revoke \Leftrightarrow valid_d(u,r,t) = [t_d, t_e]$，$t > t_e$，$valid_d(u,r,t)$ 是将关系 $U \times R \times T$ 映射到时间段 $[t_d, t_e]$ 的函数，表示用户 u 被代理的角色 r 在 $[t_d, t_e]$ 时间段内是有效的. 这种方式简单方便，尤其适用短时间段的角色代理. 但对于滥用代理角色的用户，这种撤销方式的及时性不好，因此系统应该还支持用户主动撤销方式[15].

用户主动撤销方式是角色代理后，由授权的用户在任何时候手动撤销. 在 RDM2000 模型中，有两种用户撤销方式：(1) 授予依赖(grant-dependent)方式；(2) 授予独立(grant-independent)方式. 在本模型中，所有的角色都是代理角色或代理角色的分解角色，所以只有授予依赖方式. 授权代理撤销的关系如下表示.

定义 8. $can_revoke \subseteq R$，如果 $(r) \in can_revoke$，则角色 r 的用户 $User(r)$ 均可以被代理路径的任意前驱用户撤销该角色的代理.

3.4　基于规则的策略规范语言

在 VORD 模型中，代理、撤销和分解策略都是嵌入到模型组件(代理、撤销和分解关系)中的. 而这些组件是用形式化方法来描述的，需要一种语言来显示地声明并执行这些策略.

RDM2000 模型提出一个基于规则的策略规范语言(Role-based Policy Specification Language，RPSL)[9]. RPSL 是一种将逻辑和规则绑定在一起的声明性语言(declarative language)，非常适合安全管理员定义策略. 在 RPSL 中，策略被表示为授权规则集合，通过对有限规则集进行计算可以得出代理授权.

RPSL 中的一个规则可以表达为

$$H \leftarrow F_1 \& F_2 \& \cdots \& F_n,$$

其中 $H, F_1, F_2, \cdots, F_n$ 是布尔函数，可读 H 为真，F_1 为真，且 F_2 为真，…，F_n 为真.

RPSL 的基本元素是一个函数集，每个函数有一个名字、一组参数和一个返回值. 一个函数可以是另一个函数的参数，返回值为真或假的也称作布尔函数. RPSL 包括三种函数：从 RDM2000 模型中映射来的函数、公用函数和授权函数. 详细函数列表请参见文献[9].

我们采用 RPSL 作为 VORD 模型的策略规范语言. 由于 VORD 引入角色分解组件，需要增加如下的函数和授权规则，以便虚拟组织使用这些规则来对角色分解操作进行授权.

从模型映射来的函数：

$decomposeOf(r, r')$ 表示如果 r 是 r' 的分解因子，则返回值为真. 这个函数将模型中的关系：$r \lhd r'$ 映射成系统函数.

$cardinality(r)$ 表示角色 r 代理树的路径和，即角色 r 的代理势.

公用函数：

$decomposable(u, r)$ 表示如果用户 u 有权分解角色 r，则返回值为真.

内嵌在模型内的基本授权规则：

$can_decompose(r, cr) \leftarrow$. r 是角色，cr 是先决条件.

这条规则是从上述分解关系 $can_decompose \subseteq R \times CR$ 抽取出的分解规则，表示如果用户是 r(或者 r 的高级角色)的成员，且满足条件 cr，则可以分解角色 r.

用于策略执行的授权推导规则：

$der_can_decompose(u, r, cr) \leftarrow can_decompose(r, cr) \& active(u, r', s) \& in(u, cr) \& seniorEQ(r', r) \& decomposable(u, r')$，其中 u 是用户，r 和 r' 是角色，cr 是先决条件.

这条推导规则表示，r 的高级角色 r' 的成员用户 u，如果满足先决条件 cr，则有权分解角色 r.

3.5　原型实现与评价

我们在某市的低成本电子政务网格试验平台中，实现了本模型的原型系统. 此试验平台是给该机关提供一个动态的资源共享和协作政务流程的虚拟空间. 每个委办局(共享资源的自治域)的办公自动化信息系统采用的都是基于角色的访问控制系统.

该试验平台权限管理的体系结构如图 4 所示. 其中，代理管理器集成了 Permis 系统中的属性证书管理器 ACM 和代理发布服务 DIS①. 委办局的代理管理器通过发布代理角色属性证书来实现角色的代

① http://sec.cs.kent.ac.uk/permis/private/wip.html#ACM

理功能. 虚拟空间的代理角色管理器抽取出属性证书中的本地角色和相关信息形成代理角色,存储到虚拟空间的角色库. 虚拟角色和分解角色管理器依据虚拟空间的继承或分解授权策略,对代理角色进行继承或分解,形成虚拟角色或分解角色. 本地访问拦截器将虚拟组织用户的访问请求转换成本地请求格式,并转发给本地访问控制器来进行判定.

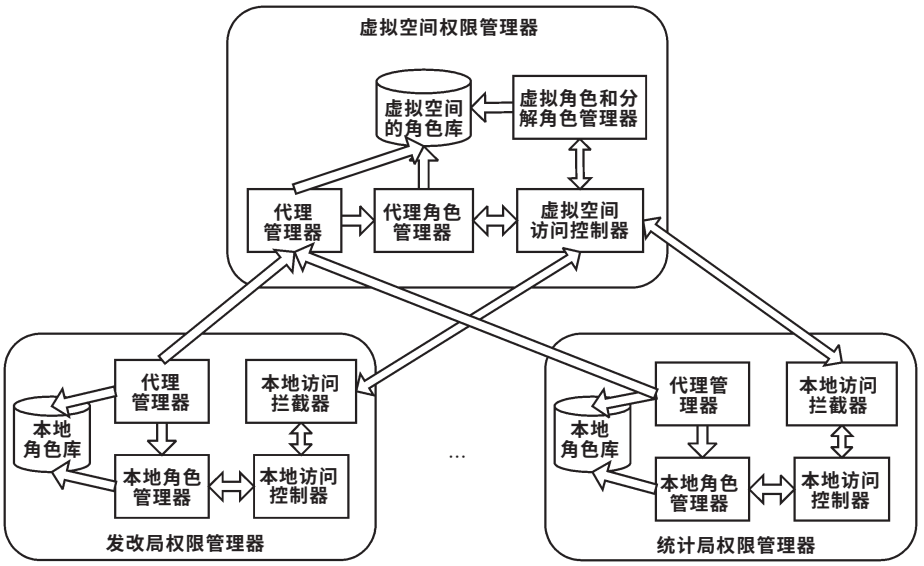

图 4 权限管理体系结构

下面以经济普查工作中的一个实际场景来分析 VORD 模型在实际应用中的效果. 每隔数年的经济普查工作需要市政府、统计局和发改局等多个部门协同完成. 为此,我们构建了一个经济普查虚拟空间 VS_{EI},相关部门将自己的资源以服务的形式共享,临时抽调的工作人员使用共享的服务相互协作来完成经济普查工作.

统计局原有的工业统计工作包括:制定工业产品生产及销售月报、工业企业主要经济指标年报和工业企业经济计划指标等报表统计工作. 为防止造假,统计局权限系统将这 3 个报表的写和读权限分别分配给 3 个角色,如表 1 所示. 并且设定授权策略规则 $Rule_1$:互斥角色集合$\{R_{PS}, R_{EI}, R_{EPI}\}$的互斥值为 1.

表 1 权限分配表

序号	角色名称	分配的权限
1	R_{PS}	PW_{PS}(写生产和销售报表) PR_{PS}(读生产和销售报表)
2	R_{EI}	PW_{EI}(写经济指标报表) PR_{EPI}(读经济指标报表)
3	R_{EPI}	PW_{EPI}(写经济计划指标报表) PR_{EPI}(读经济计划指标报表)

当统计局参与到经济普查虚拟空间 VS_{EI},并将上述 3 个报表统计封装成业务服务 BS_{PS}, BS_{EI} 和 BS_{EPI} 发布后,只需使用 ACM 向虚拟空间 VS_{EI} 签发一个如图 5 所示的代理角色属性证书 $AC_1$①,便完成了授权. 虚拟空间动态地为用户分配角色时,统计局端不需要做任何工作.

Holder: /CN=VSEI/O=Zhaoqing/C=CHN
Issuer: /CN=Tongjiju/O=Zhaoqing/C=CHN
Attributes: /DelegatedRoles=product_sale::
economy_index::economy_planning_index
ValidityFrom: 13/7/2005 1:00
ValidityTo: 14/7/2005 23:00
Signature: (签名算法)
ObjectDigest: (本证书的摘要比特流)

图 5 属性证书 AC_1

发改局的用户 U_{FGJ} 要制定下年度的经济指标计划,需要拥有读业务服务 BS_{EPI} 的权限. 虚拟空间管理员首先利用角色分解管理器创建一个 R_{EPI} 的分解角色 R_P,并将权限 PR_{EPI} 分配给 R_P;然后签发如图 6 所示的代理角色属性证书 AC_2. 用户 U_{FGJ} 只能读取业务服务 BS_{EPI},如果写业务服务 BS_{EPI},则虚拟组织访问控制器会根据角色 R_P 的权限,返回给本地拦截器拒绝该用户的访问请求.

在安全保障方面,如果虚拟空间的安全管理员错误地将角色 R_{PS} 和 R_{EPI} 代理给同一个用户,比如为用户 U_1 签发如图 7 所示的属性证书 AC_3,若 U_1 访问服务 R_{EI} 并出示该证书时,统计局的本地访问拦截器抽取出证书中的角色 R_{EI} 和 R_{EPI} 并将其转发给

① 代理角色属性证书的简化示意.

本地访问控制器后，访问控制器发现该用户角色分配关系与本地授权策略规则 $Rule_1$ 矛盾，就会拒绝 U_{1VS} 的访问请求，从而维护了统计局内部的访问控制策略.

```
Holder: /CN=U1/O=Zhaoqing/C=CHN
Issuer: /CN=VSEI/O=Zhaoqing/C=CHN
Attributes: /DelegatedRoles=rp

    Holder: /CN=VSEI/O=Zhaoqing/C=CHN
    Issuer: /CN=Tongjiju/O=Zhaoqing/C=CHN
    Attributes: /DelegatedRoles=product_sale::
                economy_index::economy_planning_index
    ValidityFrom: 13/7/2005 1:00
    ValidityTo: 14/7/2005 23:00
    Signature: (签名算法)
    ObjectDigest: (本证书的摘要比特流)

ValidityFrom: 13/7/2005 3:00
ValidityTo: 14/7/2005 23:00
Signature: (签名算法)
ObjectDigest: (本证书的摘要比特流)
```

图 6　属性证书 AC_2

```
Holder: /CN=U1/O=Zhaoqing/C=CHN
Issuer: /CN=VSEI/O=Zhaoqing/C=CHN
Attributes: /DelegatedRoles=product_sale::economy_index

    Holder: /CN=VSEI/O=Zhaoqing/C=CHN
    Issuer: /CN=Tongjiju/O=Zhaoqing/C=CHN
    Attributes: /DelegatedRoles=product_sale::
                economy_index::economy_planning_index
    ValidityFrom: 13/7/2005 1:00
    ValidityTo: 14/7/2005 23:00
    Signature: (签名算法)
    ObjectDigest: (本证书的摘要比特流)

ValidityFrom: 13/7/2005 3:00
ValidityTo: 14/7/2005 23:00
Signature: (签名算法)
ObjectDigest: (本证书的摘要比特流)
```

图 7　属性证书 AC_3

如果使用传统的用户标识映射方式，统计局需要创建 3 个统计局本地用户 U_{L1}，U_{L2}，U_{L3}，并分别分配角色 R_{PS}，R_{EI} 和 R_{EPI} 给它们，然后将 3 个本地用户映射到虚拟空间. 当虚拟空间动态地改变虚拟空间用户和本地用户的映射关系时，统计局需要随时从虚拟空间下载最新的 gridmap-file. 当互斥规则和互斥角色集合的元素很多的时候，安全管理员的授权操作步骤明显多于角色代理方式，而且还要管理 gridmap-file 文件和维护其一致性，管理工作变得非常繁琐.

通过低成本电子政务网格项目的验证和分析，VORD 模型与传统的基于身份标识的虚拟组织访问控制模型相比，在支持虚拟组织细粒度的访问控制的同时，自治域的安全管理工作变得简单，并且确保自治域的访问控制策略不被破坏.

4　总　结

服务网格环境下的虚拟组织具有动态性和自主性，给由采用 RBAC 自治域组成的虚拟组织访问控制带来新的问题，采用现有的虚拟组织访问控制模型会加重自治域的安全管理工作和给自治域带来安全漏洞. 现有的角色代理模型 RDM2000 等，因为粗粒度或单域集中管理的原因，不适用跨自治域的虚拟组织访问控制.

本文提出的 VORD 模型中的分解角色符合角色的定义，可以被继承和分解，在保证概念一致和模型原有关系不变的情况下，实现了细粒度的访问控制，从而使得虚拟组织既可以通过继承关系实现大粒度的授权，也可以通过分解关系实现细粒度的授权. 在角色代理的度上，既能对深度和广度进行约束，又能约束代理的总次数(势)，提高了模型的代理策略表达能力. 虚拟组织通过代理自治域的本地角色给用户来进行授权，用户在访问资源时，自治域的访问控制策略仍然有效，确保了自治域的安全策略不被破坏.

下一步的工作将从以下两方面来开展：

(1) VORD 模型是为了确保采用 RBAC 自治域的安全策略不被破坏而提出的，对于采用 DAC 或 MAC 的自治域，需要使用 Sylvia 等[16] 的方法对模型进行扩展，即通过增加配置角色的继承关系和约束来表达 DAC/MAC 的策略.

(2) VORD 模型是用户到用户的代理模型，为了支持更加灵活高效的代理，需要增加角色和角色之间的代理关系和相应的代理授权操作.

参　考　文　献

1　Foster I., Kesselman C., Tuecke S.. The anatomy of the grid. International Journal of High performance Computing Applications, 2001, 15(3): 200～222

2　Yu Jian, Han Yan-Bo. Service-Oriented Computing: Principles and Applications. Beijing: Thinghua University Press, 2006(in Chinese)
(俞　坚，韩燕波. 面向服务的计算——原理及应用. 北京：清华大学出版社，2006)

3　Han Y., Geng H., Li H. *et al*. VINCA—A visual and personalized business-level composition language for chaining Web-based services. In: Proceedings of the 1st International Conference on Service-Oriented Computing, Trento, Italy,

2003, 165～177

4 Foster I., Kesselman C., Pearlman L., Tuecke S., Welch V.. The community authorization service: Status and future. In: Proceedings of the Computing in High Energy Physics, La Jolla, California, USA, 2003

5 Alferi R., Cecchini R., Ciaschini V. *et al*. From gridmap-file to VOMS: Managing authorization in a grid environment. Future Generation Computer Systems Journal, 2005, 21(4): 549～558

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

7 David F. Ferraiolo *et al*. Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security, 2001, 4(3): 224～274

8 Barka E., Sandhu R.. Framework for role-based delegation model. In: Proceedings of the 23rd National Information Systems Security Conference, Baltimore, MD, 2000, 101～114

9 Zhang Long-Hua, Ahn Gail-Joon, Chu Bei-Tseng. A rule-based framework for role-based delegation. In: Proceedings of the 6th ACM Symposium on Access Control Models and Technologies, Chantilly, Virginia, USA, 2001, 153～162

10 Zhang Xin-Wen, Oh Sejong, Sandhu R.. PBDM: A flexible delegation model in RBAC. In: Proceedings of the 8th Symposium on Access Control Models and Technologies, Como, Italy, 2003, 149～157

11 Stoupa K., Vakali A., Li Fang, Tsoukalas I.. XML-based revocation and delegation in a distributed environment. In: Proceedings of the EDBT International Workshop on Database Technologies for Handling XML information on the Web, Heraklion, Greece, 2004, 299～308

12 Zhao Qing-Song, Sun Yu-Fang, Sun Bo. RPRDM: A repeated-and-part-role-based delegation model. Journal of Computer Research and Development, 2003, 40(2): 221～227(in Chinese)
(**赵庆松,孙玉芳,孙 波**. RPRDM——**基于重复和部分角色的转授权模型**. **计算机研究与发展**, 2003, 40(2): 221～227)

13 Lee Hyung-Hyo, Lee Young-Rok, Noh Bong-Ham. A new role-based delegation model using sub-role Hierarchies. In: Proceedings of the 18th International Symposium on Computer and Information Sciences, Antalya, Turkey, 2003, 811～818

14 Sandhu R., Bhamidipati V., Munawer Q.. The ARBAC97 model for role-based administration of roles. ACM Transactions on Information and System Security, 1999, 2(1): 105～135

15 Sun Bo, Zhao Qing-Song, Sun Yu-Fang. TRDM — Temporal role-based delegation model. Journal of Computer Research and Development, 2004, 41(7): 1104～1109(in Chinese)
(**孙 波,赵庆松,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**, 2004, 41(7): 1104～1109)

16 Sylvia O., Ravi S. *et al*. Configuring role-based access control to enforce mandatory and discretionary access control policies. ACM Transactions on Information and Systems Security, 2000, 3(2): 85～106

SUN Wei-Qun, born in 1975, Ph.D. candidate. His current research interests include authorization and privilege management of service grid.

SHAN Bao-Hua, born in 1978, Ph. D. candidate. His current research interests include software integration, business-level programming and service virtual organization modeling.

ZHANG Cheng, born in 1977, Ph. D. candidate. His current research interests include dynamic services composition and context-aware computing.

LIU Chen, born in 1980, Ph. D. candidate. His current research interests include service grid and software integration, service semantics.

Background

The authors are members of Sino-German Joint Laboratory of Software Integration Technologies group of Institute of Computing Technology, Chinese Academy of Sciences. The major research interests of this group are software integration, dynamic services composition and service grid. This paper is primarily supported by the National Natural Science Foundation of China (NSFC) under grant No. 90412005, which involves in the research works about virtual organization access control under service grid. This paper presents a role-based delegation access control model for virtual organizations, which named as VORD. While applied to virtual organizations, this model has little effect on security administration efficiency, can delegate partial permissions of role, and does not violate security policy of autonomous domains.