第35卷 第10期 计 算 机 工 程 2009年5月
Vol.35 No.10 Computer Engineering May 2009
·安全技术· 文章编号：1000—3428(2009)10—0155—03 文献标识码：A 中图分类号：N945.12

面向组织结构的访问控制模型

赵小龙[1]，张毓森[1]，袁 峰[2]
(1. 解放军理工大学指挥自动化学院，南京 210007；2. 国家信息安全工程技术研究中心，北京 100093)

摘 要：引入组织域的概念，描述企业组织的层状结构，在此基础上重新定义访问控制要素，提出面向组织结构的访问控制(OSOAC)模型，并扩展得到等级OSOAC模型和约束OSOAC模型。与RBAC模型相比，OSOAC模型能减少角色数量和权限分配关系，降低大型访问控制系统的管理复杂度。

关键词：访问控制；组织域；角色；等级；约束

Organization-Structure Oriented Access Control Model

ZHAO Xiao-long[1], ZHANG Yu-sen[1], YUAN Feng[2]
(1. School of Command Automation, PLA University of Science and Technology, Nanjing 210007;
2. National Information Security Engineering Technology Research Center, Beijing 100093)

【Abstract】The concept of organization domain is introduced to describe the hierarchical structure of the enterprise organization. Based on the concept, the elements of access control are redefined and an Organization-Structure Oriented Access Control(OSOAC) model is proposed. The hierarchical OSOAC model and constrained OSOAC model are drawn by extended the Core OSOAC model. Contrast to RBAC model, there are fewer roles and permission assignment relations in OSOAC model, which reduce the privilege-management complexity in a large access control system.

【Key words】access control; organization domain; role; hierarchy; constraint

1 概述

访问控制技术是显式的准许或限制访问能力及范围的一种方法，是信息安全技术中针对越权使用资源的防御措施。当前最为热点的访问控制技术是RBAC[1]，其思想是在用户和权限之间引入角色，为用户指派适当的角色，为角色分配不同的权限，用户通过激活的角色行使相应的权限。在RBAC的应用模型中RBAC96[2]得到了广泛认可，它包括4个子模型：$RBAC_0$定义了用户，角色，权限，会话的集合以及用户角色指派、角色权限分配关系，$RBAC_1$在$RBAC_0$的基础上增加了角色继承，$RBAC_2$在$RBAC_0$基础上增加了限制约束，$RBAC_3$模型是对$RBAC_1$和$RBAC_2$的综合。NIST在RBAC96基础上制订了统一的RBAC标准[3]。

RBAC模型避免了直接配置主体与客体的访问控制关系，但在大型系统的工程应用中，随着系统规模的扩大，角色数量急骤增加，与客体、操作的关联关系成倍数增长，权限管理工作仍然过于庞大。针对此问题，本文根据企业组织构成的实际特点，提出一种面向组织结构的访问控制(Organization-Structure Oriented Access Control, OSOAC)模型，将角色与系统的组织结构相互关联，进一步降低了权限管理的复杂度。

2 组织域的引入

2.1 实例分析

现代企业的一个重要特征是按层状结构进行组织管理，图1是某企业的组织结构，自上而下逐层涵盖了企业的各个分支机构和部门。在实际工作中，企业的所有员工均在公司的某级组织内担任角色行使职责，而公司的内部的各种资料设备则分布于各级组织，由组织内具有相应权限的员工进行各种操作。

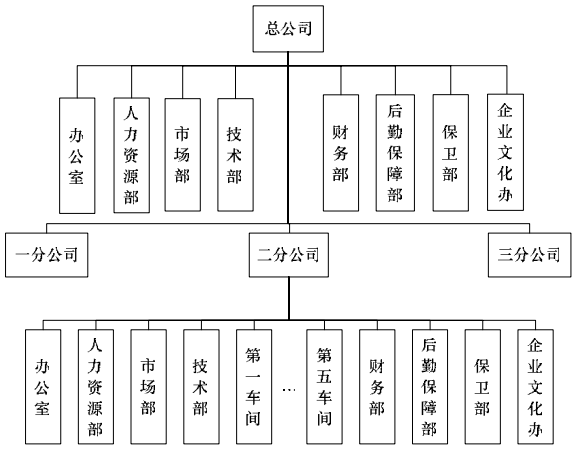

图1 某企业组织结构

由此可见，组织单元不仅是企业对人员进行管理配置的基本单位，也是企业资源的操作权限如何分配的构成依据。在$RBAC_1$模型中，角色等级的定义部分涵盖了对组织结构的处理，如系统可配置总公司经理和一分公司经理、二分公司经理、三分公司经理这样4个角色分别授予权限，而总公司经理为其余3个角色的上级角色从而继承它们的权限，但这

作者简介：赵小龙(1978－)，男，博士研究生，主研方向：信息安全，应用密码学；张毓森，教授、博士生导师；袁 峰，工程师
收稿日期：2008-09-23 **E-mail：**aolng@126.com

样的处理显得过于冗余。

从组织结构的角度来分析，以上 4 个在 RBAC 模型下的不同角色具有 2 个相同的特点：(1)对本组织单元内的资源具有相同的访问控制权限；(2)与本组织单元内的其他角色具有相同的等级关系。另外，加入组织单元的限定后，根据组织单元的层次关系可清晰辨别这 4 个角色相互间的上下等级关系。实际上，如果考虑组织单元，这 4 个角色可简化为同 1 个角色——经理来进行处理，如图 2 所示。

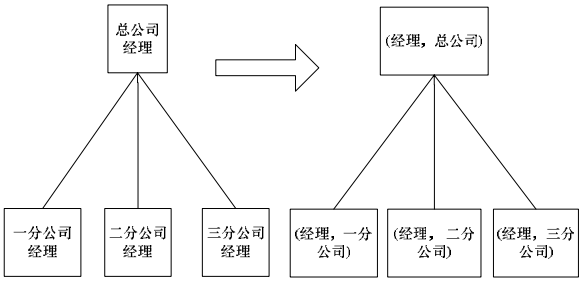

图 2　组织单元对角色的简化

从该实例可见，组织结构是决定企业访问控制策略的 1 个重要因素，如加以考虑，可极大地减少 RBAC 模型的角色数量，从而降低访问控制系统的管理复杂度。

2.2　面向组织结构的访问控制策略

根据企业环境下组织结构的一般职能，结合企业部门中用户角色行使职责的实际情况，本文提出一种面向组织结构的访问控制策略，基本思路如下：

(1)用户分配的角色是隶属于某级组织单元的，组织单元限定了用户角色行使操作权限的范围。

(2)客体在组织单元的基础上进行分类，如果客体的某种操作权限为某级组织单元内所有具备该操作权限的角色共享，则将这些客体归入一种客体类型，理解为都是可在该组织单元内执行该操作的客体。

(3)权限将角色与分类后的客体关联起来，定义为角色对某一类客体的操作能力。

(4)根据以下 2 个条件判断用户能否对客体进行某种操作：1)用户所分配的角色是否具备执行此操作的权限；2)用户所分配的角色是否处在客体该操作所隶属的组织单元内。只有当以上 2 个条件都满足时用户方可对客体进行操作，否则用户对客体的操作将被拒绝。

2.3　组织结构描述

从访问控制的角度来看，企业组织单元实际上是一个具有层状结构的主体与客体的集合，这个集合内的客体具有某种操作权限为这个集合内指定此操作的主体所共享，且不能为该组织单元外的其他主体所享有。由此得到组织域的定义。

定义 1(组织域) 组织域是访问控制系统中主体与客体集合的逻辑边界，这个逻辑边界内，主体对客体可具有某种排它性的操作权限。用 *ODS* 表示所有组织域的集合。组织域的层次结构如图 3 所示。

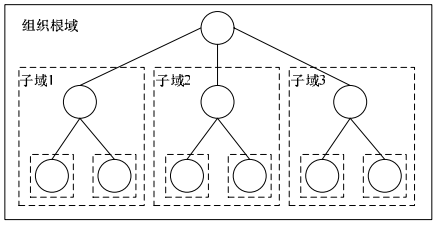

图 3　组织域的层次结构

组织域具有层状结构，整个系统范围是组织根域，每个下级组织单元为组织子域，分层结构构成了组织域树，如图 3 所示。每个组织域具有 0 个或多个子域，0 个或 1 个父域，没有父域的组织域是组织根域，每个系统有且只有一个组织根域。由于组织域的层次关系满足自反性、传递性和非对称性，因此是一个偏序关系，可形式化描述为

(1) $DH \subseteq ODS \times ODS$ ，表示集合 *ODS* 上的偏序关系，称为组织域的层次关系，记作 $\geq_{DH}$ 。$d_1 >_{DH} d_2$ 表示 d_2 是 d_1 的子域，$d_1 \geq_{DH} d_2$ 表示 $d_1 >_{DH} d_2 \vee d_1 = d_2$ ，$d_1 \gg_{DH} d_2$ 表示 d_2 是 d_1 的直接子域。

(2) $\forall d_1, d_2, d_3 \in ODS$ ，则 $d_2 \gg_{DH} d_1 \wedge d_3 \gg_{DH} d_1$ 。

(3) $\Rightarrow d_2 = d_3$ ，每个组织域最多只能有一个直接父域。

3　OSOAC 模型

以组织域的概念为基础，本节重新定义了访问控制要素，对面向组织结构的访问控制策略进行形式化描述，提出 OSOAC 模型，如图 4 所示。

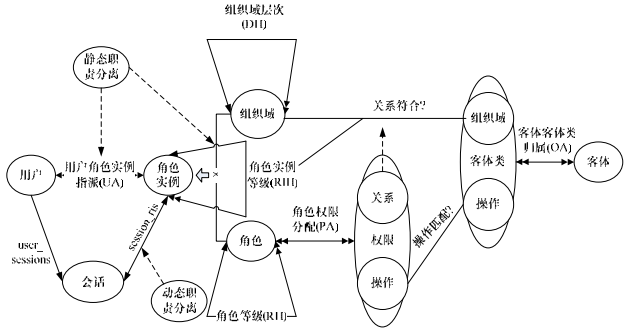

图 4　OSOAC 模型

3.1　基本 OSOAC 模型

基本 OSOAC 模型包括基本元素及其映射关系集合，思路是将用户指派到角色实例，将对象归属到对象类，将权限分配给角色，角色实例、客体类之间通过权限构成多对多的映射关系，用户由此得到对客体的操作能力。基本 OSOAC 模型也定义了会话的集合，会话是用户到指派给他的角色实例集合中被激活子集的映射。

在 OSOAC 模型中，角色不与具体的客体操作直接关联，角色只有与组织域结合起来才能明确它的权限范围，分配给用户的是隶属于组织域的角色实例，其定义为如下：

定义 2(角色实例) 二元组 (r, od) 定义为角色实例，其中，r 是角色名称；od 是角色隶属组织域的名称；集合 *RIS* 表示所有角色实例的集合。

客体根据其操作的不同最大共享范围划归到不同的客体类，单个客体允许执行操作的数量决定了其应归属的客体类的数量，用组织域描述客体的最大共享范围，其定义为

定义 3(客体类) 二元组 (op, od) 定义为客体类，其中，od 是客体类隶属组织域的名称；op 是组织域内分配此权限的主体对该类客体可执行操作的名称；集合 *OBCS* 表示所有客体类的集合。

角色实例与客体类之间通过权限进行关联，OSOAC 模型中的权限是主体对满足条件的客体执行操作的核准。权限分配给角色，但角色只有与组织域结合为具体的角色实例后才能明确权限的作用范围，权限中要求满足的条件用于限定角色实例 (r, od_1) 和客体类 (op, od_2) 所隶属的组织域之间的关系，分为 $od_2 = od_1$ 和 $od_2 \geq_{DH} od_1 2$ 种情况，$od_2 = od_1$ 表示

角色实例仅对隶属于本组织域此客体类执行该操作，而 $od_2 \geq_{DH} od_1$ 表示角色实例可对隶属于本组织域及父域的此客体类执行该操作。权限的定义为

定义 4(权限) 二元组 (op, rel) 定义为权限，其中，op 是赋予角色的操作的名称；rel 是角色实例和客体类所隶属的组织域之间的关系；集合 $PRMS$ 表示所有权限的集合。

下文对基本 OSOAC 模型进行形式化描述，其中，$ODS, USERS, ROLES, OPS, OBS$ 分别表示组织域、用户、角色、操作、客体的集合。

(1) $RIS = 2^{(ROLES \times ODS)}$，角色实例的集合。

$UA \subseteq USERS \times RIS$，多对多映射的用户角色实例指派关系。

$assigned\_users(ri : RIS) \to 2^{USERS}$，角色实例 ri 到用户集合的映射，形式化描述为

$$assigned\_users(ri) = \{u \in USERS \mid (u,\ ri) \in UA\}$$

$ris\_domain(ri : RIS) \to ODS$，角色实例到其隶属组织域的映射。

(2) $OBCS = 2^{(OPS \times ODS)}$，客体类的集合。

$OA \subseteq OBS \times OBCS$，多对多映射的客体客体类归属关系。

$attached\_objs(obc : OBCS) \to 2^{OBS}$，客体类 obc 到客体集合的映射，形式化描述为

$$attached\_objs(obc) = \{obj \in OBS \mid (obj,\ obc) \in OA\}$$

$obc\_domain(obc : OBCS) \to ODS$，客体类到其隶属组织域的映射。

(3) $RELS$，角色实例隶属组织域和客体类隶属组织域之间关系的集合。

$PRMS = 2^{(OPS \times RELS)}$，权限的集合。

$PA \subseteq PRMS \times ROLES$，多对多映射的权限角色分配关系。

$authorized\_permissions : (r : ROLES) \to 2^{PRMS}$，角色 r 到权限集合的映射，形式化描述为

$$authorized\_permissions(r) = \{p \in PRMS \mid (p,\ r) \in PA\}$$

(4) $SESSIONS$，会话的集合。

$user\_sessions(u : USERS) \to 2^{SESSIONS}$，用户 u 到会话集合的映射。

$session\_users(s : SESSIONS) \to 2^{USERS}$，会话 s 到用户集合的映射。

$session\_ris(s : SESSIONS) \to 2^{RIS}$，会话 s 到角色实例集合的映射，形式化描述为

$$session\_ris(s) = \{ri \in RIS \mid (session\_users(s),\ ri) \in UA\}$$

$avail\_session\_perms(s : SESSIONS) \to 2^{PRMS}$，会话 s 到会话中所分配权限集合的映射，该权限表述为

$$\bigcup_{ri \in sessoin\_ris(s)} assigned\_permissions(ri)$$

(5)访问规则为

$\forall u : USERS, op : OPS, obj : OBS,\ access : USERS \times OPS \times OBS \to BOOLEAN$，如果 u 可对 obj 执行 op，则 $access(u,\ op,\ obj) = 1$，否则，$access(u,\ op,\ obj) = 0$。

$access(u,\ op,\ obj) = 1 \Rightarrow \exists ri \in RIS,\ obc \in OBCS,\ \ prm \in PRMS,\ s \in SESSIONS$，满足 $u \in session\_(u, ri) \in UA \wedge (obj, obc) \in users(s) \wedge OA \wedge (prm, ri.r) \in PA$，且 $prm.op = obc.op \wedge obc.dom\ \ prm.rel\ \ ri.dom$。

3.2 等级 OSOAC 模型

等级 OSOAC 模型定义了角色实例等级关系以反映企业组织环境下的职责分布和权限继承关系，角色实例等级关系是在 OSOAC 模型下角色等级关系和组织域层次关系的基础上定义的二元偏序关系。

3.2.1 角色等级关系

OSOAC 模型下角色等级关系用于反应角色权限的可继承性，如果角色 r_2 的所有权限都是角色 r_1 的权限，那么称角色 r_1 继承角色 r_2。在角色等级关系中，上级角色继承下级角色的权限，每个角色可具有 0 个或多个直接上级角色，也可具有 0 个或多个直接下级角色。角色等级关系满足自反性、传递性和非对称性，是一个偏序关系。

3.2.2 角色实例等级关系

角色实例是角色和组织域的笛卡尔积，由于角色等级关系和组织域层次关系为偏序关系，在此基础上定义角色实例等级关系也满足自反性、传递性和非对称性，是一个偏序关系。形式化描述为

$RIH \subseteq RIS \times RIS$，集合 RIS 上的偏序关系，称为角色实例等级关系，记作 $\geq_{RH}$。$ri_1 >_{RH} ri_2$ 表示 ri_1 是 ri_2 的上级角色，$ri_1 \geq_{RH} ri_2$ 表示 $ri_1 >_{RH} ri_2 \vee ri_1 = ri_2$。

$$ri_1 \geq_{RH} ri_2 \Rightarrow authorized\_permissions(ri_2.r) \subseteq authorized\_permissions(ri_1.r) \wedge\ assigned\_users(ri_1) \subseteq assigned\_users(ri_2)$$

$assigned\_users : (ri : RIS) \to 2^{USERS}$，角色实例等级关系下角色实例 ri 到权限集合的映射，形式化描述为

$$assigned\_users(ri) = \{u \in USERS \mid ri' \geq_{RH} ri,\ \ (u, ri') \in UA\}$$。

引入角色实例等级关系后，访问规则形式化描述为

$access(u, op, obj) = 1 \Rightarrow$

$\exists ri, ri' \in RIS, obc \in OBCS, prm \in PRMS, s \in\ SESSIONS$，满足 $u \in session\_users(s) \wedge (u,\ ri) \in UA \wedge ri \geq_{RH} ri' \wedge (obj,\ obc) \in OA \wedge PA$，且 $prm.op = obc.op \wedge\ obc.dom\ \ prm.rel\ \ ri'.dom$。

3.3 约束 OSOAC 模型

约束 OSOAC 模型定义了 OSOAC 模型下的职责分离关系，以支持访问控制模型的“最小权限原则”[4]。

3.3.1 静态职责分离

静态职责分离(SSD)限定用户角色实例指派关系，依据 SSD 策略，指派了一个角色实例的用户不能指派为与现有职责发生冲突的其它角色实例，在 OSOAC 模型中定义了相互排斥的组织域来满足这一约束原则，如果两个或多个组织域是相互排斥的，那么用户不能在这些组织域内指派角色实例。对组织域 SSD 策略的形式化描述为

如果 od_1 和 od_2 是相互排斥的组织域，那么这 2 个组织域不能有指派给同一用户的角色实例，即 $SSD \subseteq 2^{ODS} \times N$，SSD 策略 (ds, n) 的集合，ds 是一个组织域的集合，n 为大于 1 的自然数，表示任意用户不能在 n 个或更多的 ds 集合元素内指派角色实例，则

$$\forall (ds, n) \in SSD, \forall t \subseteq ds, |t| \geq n \Rightarrow \bigcap_{ri.od \in t} assigned\_users(ri) = \varnothing$$

3.3.2 动态职责分离

动态职责分离(DSD)限定用户在会话中激活的角色实例，由于支持用户在不同时间具有不同级别的权限，DSD 为最小权限原则提供了更大的灵活性。OSOAC 模型定义了不能同时激活的组织域来满足该约束原则。

如果 od_1 和 od_2 是相互排斥的组织域，那么在任一个会话中不能同时激活指派给同一用户的这 2 个组织域内的角色实例，形式化描述为

(下转第 161 页)