基于角色的时态对象存取控制模型

王小明[1], 赵宗涛[2]
(1. 陕西师范大学计算机科学学院, 陕西西安 710062; 2. 第二炮兵工程学院计算机科学系, 陕西西安 710025)

摘 要: 以基于角色的存取控制模型$RBAC_3$为基础, 提出一种新的时态对象存取控制模型TRBAC, 讨论了模型的构成要素, 体系结构, 时态多重继承机制和存取控制方法. 它支持时态用户、时态角色和时态客体及其层次结构, 具有动态存取控制功能, 在时态数据库和工作流系统等领域会得到广泛应用.

关键词: 时态对象; 存取控制; 角色; 时间约束; 时态继承机制
中图分类号: TP309 文献标识码: A 文章编号: 0372-2112 (2005) 09-1634-05

Role-Based Access Control Model of Temporal Object

WANG Xiao-ming[1], ZHAO Zong-tao[2]
(1. *College of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi 710062, China;*
2. *Department of Computer Science, The Second Artillery Engineering College, Xi'an, Shaanxi 710025, China*)

Abstract: A novel temporal object access control model (TRBAC) is proposed based on the role-based access control model ($RBAC_3$). The elements, architecture, temporal inherent mechanism and access control method of TRBAC are discussed respectively. It supports temporal user, temporal role and temporal object and their hierarchy structures, and has the dynamic access control power. It may have applications in such as temporal database and workflow management system.

Key words: temporal object; access control; role; time constraint; temporal inherence mechanism

1 引言

存取控制模型通过存取授权机制实现信息系统的存取控制安全策略, 它是信息安全领域研究的重点之一[1]. 目前的存取控制模型主要有自主存取控制(DAC, Descritionary Access Control), 强制存取控制(MAC, Mandatory Access Control)和基角色的存取控制(RBAC, Role-Based Access Control). RBAC模型以高效的授权管理, 面向应用层的自然映象等特点, 近年来其理论研究和应用开发发展迅速, 被公认为当前最有发展潜力的新一代存取控制模型, 正得到深入研究和广泛应用[1]. 但是, 现有的RBAC模型对时间约束的支持功能还相当简单, 因此它对时态对象的存取控制建模能力弱. 然而, 时间因素在绝大多数信息系统中无处不在, 迫切需要RBAC模型能够支持复杂的时间约束建模.

文献[2]提出的时间约束存取控制模型仅适合对用户-权限授权策略建模, 不适合RBAC策略. 文献[3, 4]对角色授权约束进行了深入研究, 但未涉及时间约束. 目前, 从时间维度对RBAC进行研究的文献还很少. 文献[5]虽然提出了带时间约束的RBAC模型, 但是把时间约束与其他约束存储在同一个约束规则库中, 不容易实现角色层次结构上的动态时态继承性, 而且效率低. 为此, 本文以$RBAC_3$模型为基础, 提出一种新的时态RBAC模型, 简称为TRBAC, 它把模型要素及其关系上的时间约束嵌入到模型之中, 通过定义新的时态继承机制实现动态基于角色的存取控制, 并且能够有效减少约束规则库中的规则数量, 从而提高存取控制效率. 本文主要讨论TRBAC模型的构成要素、体系结构和时态继承机制. 模型的其他约束与$RBAC_3$模型类似, 详细内容请参阅文献[1, 3], 限于篇幅, 本文不再详述.

2 TRBAC模型

TRBAC模型的基本思想是对$RBAC_3$模型[1]的用户、角色及其层次结构、许可和客体进行时态化扩展, 并且建立时态客体层次结构, 从而对存取控制诸要素实施静态有效性时间约束; 在存取控制决策过程中, 根据决策要素之间的时态关系和角色层次结构上的时态继承机制实施动态时间约束.

2.1 时间表示

时间的有效表示是支持时间约束的存取控制模型的关键. 时间包括连续时间和离散时间两种[6]. 为简单起见, 本文以离散时间为讨论对象, 连续时间讨论与此类似. 设$T=\{t_0, t_1, \cdots\}$是离散时间轴上的所有点构成的无限集合. 称离散时

收稿日期: 2005-02-04; 修回日期: 2005-04-26

间轴上的不间断点构成的集合为区间时间, 记作$[t_s, t_e]$, 其中$t_s \leqslant t_e$. 区间时间全体记为 IT. 为描述统一起见, 时间点 $t \in T$ 用区间时间$[t, t]$表示.

设 max: $T \times T \to T$ 和 min: $T \times T \to T$ 分别是时间集上的取大和取小函数. 对任意区间时间$[t_{s1}, t_{e1}]$和$[t_{s2}, t_{e2}]$, 它们的交和并分别定义为:

$$[t_{x1}, t_{y1}] \cap [t_{x2}, t_{y2}] = \begin{cases} [\max(t_{x1}, t_{x2}), \min(t_{y1}, t_{y2})], \\ \qquad\qquad 当\ t_{x1} \leqslant t_{x2} \leqslant t_{y1} \\ \varnothing, \qquad 当\ t_{y1} < t_{x2} \end{cases} \quad (1)$$

$$[t_{x1}, t_{y1}] \cup [t_{x2}, t_{y2}] = \begin{cases} [\min(t_{x1}, t_{x2}), \max(t_{y1}, t_{y2})], \\ \qquad\qquad 当\ t_{x1} \leqslant t_{x2} \leqslant t_{y1} \\ [t_{x1}, t_{y1}] + [t_{x2}, t_{y2}], 当\ t_{y1} < t_{x2} \end{cases} \quad (2)$$

其中, 符号"+"表示两个区间时间同时存在. 如果 $t_{x2} \leqslant t_{x1}$并且 $t_{y1} \leqslant t_{y2}$, 则称$[t_{x2}, t_{y2}]$包含$[t_{x1}, t_{y1}]$, 记作$[t_{x1}, t_{y1}] \subset [t_{x2}, t_{y2}]$. 对任意 $t \in T$, 如果 $t_{x1} \leqslant t \leqslant t_{y1}$, 则称 t 在$[t_{x1}, t_{y1}]$内, 记作 $t \in [t_{x1}, t_{y1}]$.

设 M 和 L 是两个区间时间集合, M 和 L 的交集 $M \cap L$ 定义为:

$$M \cap L = \{[t_x, t_y] \mid \forall [t_x^M, t_y^M] \in M, \exists [t_x^L, t_y^L] \in L, 令 [t_x, t_y] = [t_x^M, t_y^M] \cap [t_x^L, t_y^L]\} \quad (3)$$

M 和 L 的并集$M \cup L$ 定义为:

$$M \cup L = \{[t_x, t_y] \mid \forall [t_x^M, t_y^M] \in M, \exists [t_x^L, t_y^L] \in L, 令 [t_x, t_y] = [t_x^M, t_y^M] \cup [t_x^L, t_y^L]\} \quad (4)$$

M 是 L 的子集, 记为 $M \subseteq L$, 则

$$M \subseteq L \Leftrightarrow \forall [t_x^M, t_y^M] \in M, \exists [t_x^L, t_y^L] \in L: [t_x^M, t_y^M] \subset [t_x^L, t_y^L] \quad (5)$$

2.2 时态对象

在系统安全研究中, 对象主要指主体、客体和权限等. 对象通常使用属性刻划其特质. 对象的属性一些随时间推移其值不变, 称之为常量属性(constant attribute), 一些随时间推移其值可能发生变化, 称之为时变属性(time-varying attribute). 时态对象形式定义如下:

定义 1　时态对象是一个三元组 $TVA = (Oid, Aid, TV)$, 其中 Oid 和 Aid 分别是对象唯一标识和对象的时变属性唯一标识集合, $TV = \{(t, v) \mid t \in T, v \in V\}$ 是时间与属性值的对偶集合, 其中 V 是属性值集合. 定义映射 $f: Oid \times Aid \times IT \to V$, 对任意 $\sigma_i \in Oid$, 任意 $\varepsilon_j \in Aid$, 任意$[t_x, t_y] \in IT$, 存在 $v \in V$, 使得$f(\sigma_i, \varepsilon_j, [t_x, t_y]) = v$, 即标识为 σ_i 的对象的属性ε_j 在时间$[t_x, t_y]$内取值为 v.

具有相同属性与结构的时态对象构成时态对象类(class).

2.3 时态对象存取控制策略

为了与存取控制术语一致, 以下时态对象概念具体指时态主体(subject)、时态客体(object), 时态角色(role)和时态许可(permission). 非时态对象为操作(operate). 角色之间的偏序关系构成角色层次结构, 父角色继承其子角色的许可; 客体之间的偏序关系构成客体层次结构, 父客体包含其子客体. 除 RBAC 基本存取控制策略之外, 定义普遍适合时态对象的存取控制策略如下:

策略 1　用户通过角色享有许可, 从而对客体实施操作, 当且仅当四者均在其有效时间内才被允许.

策略 2　在时态角色层次结构上, 父角色继承的子角色的许可的有效时间是相应继承路径上所有角色的有效时间的交集.

策略 3　在时态客体层次结构上, 父客体继承子客体包括时间约束在内的所有属性(值).

策略 4　如果授权某用户通过某角色存取某客体, 则该用户能够通过该角色存取该客体继承的所有子客体.

显然, $RBAC_3$ 模型不支持上述存取控制策略. 为此, 本文提出支持上述存取控制策略的安全模型 TRBAC, 它是对 $RBAC_3$ 模型的时态化扩展.

2.4 TRBAC 模型构成要素

TRBAC 模型由一组集合、关系和函数构成, 定义如下:

定义 2　时态用户 $u^t = (u, \Gamma^u)$, Γ^u 是 u 的静态有效性时间约束集. 对$\forall [t_x^u, t_y^u] \in \Gamma^u$, u 在$[t_x^u, t_y^u]$内有效. 时态用户全集记为 U^t.

定义 3　时态角色 $r^t = (r, \Gamma^r)$, Γ^r 是 r 的静态有效性时间约束集. 对$\forall [t_x^r, t_y^r] \in \Gamma^r$, r 在$[t_x^r, t_y^r]$内有效. 时态角色全集记为 R^t.

定义 4　时态角色层次结构是一个二元偏序关系 $RH^t \subseteq R^t \times R^t$. 对$\forall r_i^t, r_j^t \in R^t$, 如果$(r_i^t, r_j^t) \in RH^t$, 则称 r_i^t 是r_j^t 的父角色, r_j^t 是r_i^t 的子角色, 记作 $r_i^t \geqslant r_j^t$. 当 $r_i^t \neq r_j^t$ 时, 记作 $r_i^t > r_j^t$. 如果$(r_i^t, r_j^t) \in RH^t$, 但不存在 r_k^t, 使得 $r_i^t \geqslant r_k^t$, 并且 $r_k^t \geqslant r_j^t$ 成立, 则称 r_i^t 是r_j^t 的直接父角色, 记作 $r_i^t \geqslant\!\!\cdot\ r_j^t$.

r_i^t 时态继承 r_j^t 的许可. r_i^t 与r_j^t 之间的时间约束为:

$$\forall r_i^t, r_j^t \in R^t: (r_i^t, r_j^t) \in RH^t \Rightarrow \Gamma^{r_i} \cap \Gamma^{r_j} \neq \{\varnothing\} \quad (6)$$

定义 5　时态客体 $o^t = (o, \Gamma^o)$, Γ^o 是 o 的静态有效性时间约束集. 对$\forall [t_x^o, t_y^o] \in \Gamma^o$, o 在$[t_x^o, t_y^o]$内有效. 时态客体全集记为 O^t.

定义 6　时态客体层次结构是一个二元偏序关系 $OH^t \subseteq O^t \times O^t$. 对$\forall o_i^t, o_j^t \in O^t$, 如果$(o_i^t, o_j^t) \in OH^t$, 则称 o_i^t 是o_j^t 的父客体, o_j^t 是 o_i^t 的子客体, 记作 $o_i^t \geqslant o_j^t$. 当 $o_i^t \neq o_j^t$ 时, 记作 $o_i^t > o_j^t$. 如果$(o_i^t, o_j^t) \in OH^t$, 但不存在 o_k^t, 使得 $o_i^t \geqslant o_k^t$, 并且 $o_k^t \geqslant o_j^t$ 成立, 则称 o_i^t 是o_j^t 的直接父客体, 记作 $o_i^t \geqslant\!\!\cdot\ o_j^t$.

如果 o_i^t 时态包含 o_j^t, 则 o_i^t 与o_j^t 之间的时间约束为:

$$\forall o_i^t, o_j^t \in O^t: (o_i^t, o_j^t) \in OH^t \Rightarrow \Gamma^{o_i} \subseteq \Gamma^{o_j} \quad (7)$$

定义 7　操作是允许在时态对象上可执行的最小动作单位, 其全集记为 OP.

定义 8　时态许可是一个三元组 $p^t = (p, \Gamma^p)$. 其中, $p \subseteq OP \times O^t$, Γ^p 是p 的静态有效性时间约束集. 对$\forall [t_x^p, t_y^p] \in \Gamma^p$, p 在$[t_x^p, t_y^p]$内有效. 时态许可全集记为 P^t.

时态许可满足下列时间约束:

$$\forall p^t \in P^t, \forall (op, o^t) \in p: p^t = (op, o^t) \Rightarrow \Gamma^p \cap \Gamma^o \neq \{\varnothing\} \quad (8)$$

定义 9　会话是用户激活角色的进程, 其全集记为 $S = \{s_1, s_2, \cdots, s_n\}$. 其中 n 是非负整数.

通过赋值关系把权限分配给用户, 许可赋值给角色. 用户

通过会话激活角色, 从而使用角色包含的有效许可, 实现对客体的存取操作. 赋值关系定义如下:

定义 **10** 赋值关系

(1) 时态用户-时态角色赋值多对多关系 $URA^t \subseteq U^t \times R^t$. 对 $\forall u^t \in U^t$, $\forall r^t \in R^t$, 如果 $(u^t, r^t) \in URA^t$, 则称 u^t 是 r^t 的用户成员, 并且时态约束为:

$$\Gamma^u \cap \Gamma^r \neq \{\varnothing\} \tag{9}$$

(2) 时态角色-时态许可赋值多对多关系 $RPA^t \subseteq R^t \times P^t$. 对 $\forall r^t \in R^t$, $\forall p^t \in p^t$, 如果 $(r^t, p^t) \in RPA^t$, 则称 p^t 是 r^t 的许可成员, 并且时态约束为:

$$\Gamma^r \cap \Gamma^p \neq \{\varnothing\} \tag{10}$$

(3) 会话-时态角色激活一对多关系 $SRA^t \subseteq S \times R^t$. 对 $\forall s \in S$, $\forall r^t \in R^t$, 如果 $(s, r^t) \in SRA^t$, 则称 s 激活 r^t.

(4) 会话-时态用户激活一对一关系 $SUA^t \subseteq S \times U^t$. 对 $\forall s \in S$, $\forall u^t \in U^t$, 如果 $(s, u^t) \in SUA^t$, 则称 s 激活 u^t.

设 X 是集合, $\rho(X)$ 是 X 的幂集合. 定义下列集值函数:

定义 **11** 函数

(1) $f^{u-r}: U^t \to \rho(R^t)$, 对 $\forall u^t \in U^t$, $f^{u-r}(u^t) = \{r^t \mid (u^t, r^t) \in URA^t$, 令 $\Gamma^r_j = \Gamma^r_j \cap \Gamma^u\}$ 表示时态用户 u^t 享有的显式时态角色集合.

(2) $f^{r-u}: R^t \to \rho(U^t)$, 对 $\forall r^t \in R^t$, $f^{r-u}(r^t) = \{u^t \mid (u^t, r^t) \in URA^t$, 令 $\Gamma^u = \Gamma^u \cap \Gamma^r\}$ 表示时态角色 r^t 的显式用户集合.

(3) $f^{r-p}: R^t \to \rho(P^t)$, 对 $\forall r^t \in R^t$, $f^{r-p}(r^t) = \{p^t \mid (r^t, p^t) \in RPA^t$, 令 $\Gamma^p = \Gamma^p_j \cap \Gamma^r\}$ 表示时态角色 r^t 包含的显式时态许可集合.

(4) $f^{p-r}: P^t \to \rho(R^t)$, 对 $\forall p^t \in P^t$, $f^{p-r}(p^t) = \{r^t \mid r^t \in R^t$, $(r^t, p^t) \in RPA^t$, 令 $\Gamma^p = \Gamma^p \cap \Gamma^r\}$ 表示时态许可 p^t 隶属的显式时态角色集合.

(5) $f^{p-op}: P^t \to \rho(OP \times O^t)$, 对 $\forall p^t \in P^t$, $f^{p-op}(p^t) = \{(op \times o^t) \mid (op, o^t) \in p$, 令 $\Gamma^o = \Gamma^o \cap \Gamma^p\}$ 表示许可 p^t 包含的显式(操作, 客体)偶对集合.

(6) $f^{s-r}: S \to \rho(R^t)$, 对 $\forall s \in S$, $f^{s-r}(s) = \{r^t \mid (s, r^t) \in SRA^t\}$ 表示会话 s 激活的显式时态角色集合.

(7) $f^{s-u}: S \to U^t$, 对 $\forall s \in S$, $\exists !\ u^t \in U^t$, 使 $f^{s-u}(s) = u^t$, 表示会话 s 激活时态用户 u^t.

2.5 TRBAC 模型体系结构

TRBAC 模型体系结构与 $RBAC_3$[1] 基本相似. 不同的是 TRBAC 模型的构成要素是时态型, 而 $RBAC_3$ 模型的构成要素是非时态型. 如图 1 所示, 时态角色层次结构实例和时态客体层次结构实例分别如图 2 和图 3 所示.

在图 1~3 中, 双箭头表示多对多关系, 单箭头表示一对一关系. *CON* 是其他约束集合. 时态角色和时态客体层次结构上的继承策略是沿层次结构向上继承. 图 2 的 r^t_3 分别通过 r^t_4 和 r^t_5 继承 r^t_7 的许可, 具有时态多重继承性. r^t_3 通过 r^t_4 继承的 r^t_7 中的时态许可与 r^t_3 通过 r^t_5 继承的 r^t_7 中的时态许可可能不同, 从而形成了灵活多样的继承结果.

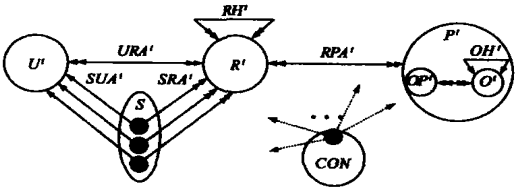

图 1 TRBAC 模型体系结构

2.6 层次结构上的时态继承机制

为讨论时态角色层次结构上的继承机制, 定义角色层次结构上的路径概念.

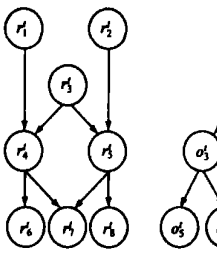

图 2 时态角色层次结构实例

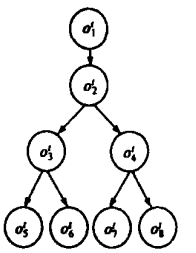

图 3 时态客体层次结构实例

定义 **12** 对 $\forall r^t_i \in R^t$, 在时态角色层次结构 RH^t 上, 以 r^t_i 为起点的路径为角色序列 $\ell = (r^t_i, \ldots, r^t_j, \ldots, r^t_m)$, 其中 $r^t_i > r^t_j$, $r^t_j \geqslant r^t_m$, 并且不存在 $r^t \in R^t$, 使得 $r^t_m > r^t$. ℓ 包含时态角色 r^t, 记作 $r^t \in \ell$. 从 r^t_i 出发的所有不同路径全集记作 $L(r^t_i)$.

r^t_i 依赖于路径 ℓ 继承的子角色(隐式角色)集合为:

$$f^{r-r}_{\ell}(r^t_i) = \{r^t \mid r^t \in \ell\} \tag{11}$$

对 $\forall r^t \in f^{r-r}_{\ell}(r^t_i)$, r^t 的有效时间集合为:

$$\Gamma^{\ell}_{r^t} = \bigcap_{\substack{r'^t \in \ell \\ r^t \geqslant r'^t}} \Gamma^{r'} \tag{12}$$

r^t_i 依赖于路径 ℓ 的有效许可集合为:

$f^{r-p}_{\ell}(r^t_i) = \{p^t \mid r^t \in f^{r-r}_{\ell}(r^t_i), (r^t, p^t) \in RPA^t$, 令 $\Gamma^p = \Gamma^{\ell}_{r^t} \cap \Gamma^p\}$ (13)

在角色层次结构 RH^t 上, r^t_i 包含的许可全集为:

$$F^{r-p}(r^t_i) = \bigcup_{\ell \in L(r^t_i)} f^{r-p}_{\ell}(r^t_i) \tag{14}$$

用符号"\"表示集合上的差运算. 在角色层次结构 RH^t 上, r^t_i 包含的有效许可全集为:

$$G^{r-p}(r^t_i) = \{p^t \mid p^t \in F^{r-p}(r^t_i), \Gamma^{\ell}_{p} \neq \{\varnothing\}\} \tag{15}$$

时态多重继承的矛盾消解可以采用"许可有效时间最大化"策略. 对 $\forall p^t \in G^{r-p}(r^t_i)$, p^t 的有效时间集合为:

$$\Gamma^p_C = \bigcup_{\ell \in L(r^t_i)} \Gamma^{p}_{\ell} \tag{16}$$

当然, 时态多继承的矛盾消解也可以采用"许可有效时间最小化"策略. 此时, 对 $\forall p^t (G^{r-p}(r^t_i), p^t$ 的有效时间集合为:

$$\Gamma^p_C = \bigcap_{\ell \in L(r^t_i)} \Gamma^{p}_{\ell} \tag{17}$$

类似地, 可以定义客体层次结构上的时态继承机制. 对 $\forall o^t_i \in O^t$, o^t_i 的子客体集合为:

$$F^{o-o}(o^t_i) = \{o^t_j \mid o^t_j \in O^t,\ o^t_i \geqslant o^t_j$, 令 $\Gamma^o_j = \Gamma^o_j\}$ (18)

2.7 时态对象存取控制算法

存取控制是当用户向系统提出存取请求时, 系统的存取控制子系统依照预先定义的存取控制策略, 在满足存取控制约束前提下, 对是否允许执行用户请求作出决策. 设 $u^t \in U^t$, $r^t \in R^t$, $t \in T$, 在时刻 t 用户存取请求表示为 $req = (u^t, r^t, [t, t])$, TRBAC 模型的存取控制算法描述如下:

Step1　如果$\{[t, t]\} \cap \Gamma^u = \{\varnothing\}$, 则转 Step10. 否则根据式(11)计算用户 u^t 当前享有的角色全集:

$$F^{u-r}(u^t) = \bigcup_{\substack{\beta \in L(r^t) \\ (u^t, r^t) \in URA^t}} f_\beta^{r-r}(r^t)$$

Step2　如果存在$(r_k, [t_x, t_y]) \in F^{u-r}(u^t)$, 使得 $r_k = r$, 并且$[t, t] \subseteq [t_x, t_y]$, 则转 Step3, 否则转 Step10.

Step3　根据式(15), 计算用户 u^t 请求执行的角色 r^t 包含的有效许可集 $G^{r-p}(r^t)$.

Step4　根据定义 11(v), 计算 $G^{r-p}(r^t)$ 中的许可包含的(操作, 客体)对偶全集:

$$OPO = \bigcup_{p^t \in G^{r-p}(r^t)} f^{p-op}(p^t)$$

Step5　对每一个$(op, o^t) \in OPO$, 根据式(18)计算 o^t 包含的子客体集 $F^{o-o}(o^t)$.

Step6　对每一个$(op, o^t) \in OPO$, 计算其扩展(操作, 客体)对偶集:

$$EOPO(op, o^t) = \{(op, o_i^t) \mid p^t \in G^{r-p}(r^t),\ o_i^t \in F^{o-o}(o^t),\ \text{令 } \Gamma^o{}_i = \Gamma^o{}_i \cap \Gamma^p\}$$

Step7　计算角色 r^t 包含的(操作, 客体) 对偶全集:

$$ROPO(r^t) = \bigcup_{(op, d) \in OPO} EOPO(op, o^t)$$

Step8　计算 u^t 能够执行的r^t 的许可集:

$$UOPO(u^t, r^t) = \{(op, o^t) \mid (op, o^t) \in ROPO(r^t),\ \text{令 } \Gamma^o = \Gamma^o \cap \Gamma^u\}$$

Step9　计算 u^t 在时刻$[t, t]$能够执行的 r^t 的许可集:

$$UOP(u^t, r^t, t) = \{(op, o^t) \mid (op, o^t) \in UOPO(u^t, r^t), [t_x, t_y] \in \Gamma^t, [t, t] \subseteq [t_x, t_y]\}$$

Step10　停止.

使用该算法可以有效实现第 2.3 节的时态对象存取控制策略 1~4.

3　应用举例

时态数据库记录如表 1, 角色层次结构如图 4.

表 1　时态数据记录

A1	D2	A2
100	d2100	(700,[92,93]) (800,[95,96]) (900,[97,98])
101	d2101	(400,[95,97]) (600,[98,99])
102	d2102	(300,[91,94]) (360,[97,98])
103	d2103	(550,[97,99])
104	d2104	(720,[97,98])

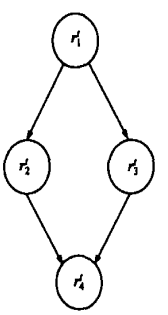

图 4　时态角色层次结构

由定义 2~7, 设:

$u_1^t = (u_1, \{[91, 94]\})$　$u_2^t = (u^2, \{[95, 97]\})$　$u_3^t = (u^3, \{[91, 96]\})$

$r_1^t = (r_1, \{[92, 94]\})$　$r_2^t = (r_2, \{[92, 96]\})$　$r_3^t = (r_3, \{[91, 98]\})$

$r_4^t = (r_4, \{[93, 95]\})$　$p_1^t = (p_1, \{[92, 98]\})$　$p_2^t = (p_2, \{[93, 94]\})$

$p_4^t = (p_4, \{[90, 95]\})$　$p_1 = (\text{mod}, A_2)$　$p_2 = (\text{pm}, A_2)$

$p_4 = (\text{del}, A_2)$

其中, mod, prn, del 分别为修改, 打印, 删除原子操作.

由定义 10, 设:

$$URA^t = \{(u_1^t, r_2^t), (u_2^t, r_3^t), (u_3^t, r_1^t)\}$$

$$RPA^t = \{(r_2^t, p_1^t), (r_3^t, p_2^t), (r_4^t, p_4^t)\}$$

当前用户存取请求为: $(u_3^t, r_1^t, [92, 92])$. 从 r_1^t 到 r_4^t 的两条路径是 $l_1 = (r_1^t, r_2^t, r_4^t)$ 和 $l_2 = (r_1^t, r_3^t, r_4^t)$. 由算法 Step1 得:

$$F^{u-r}(u_3^t) = \{(r_1, \{[92, 94]\}), (r_2, \{[92, 94]\}), (r_3, \{[92, 94]\}), (r_4, \{[93, 94]\})\}$$

显然, $r_1^t \in f^{u-r}(u_3^t)$. 由算法 Step3 和式(11)~(16)得:

$$G^{r-p}(r_1^t) = \{(p_1, \{[92, 94]\}), (p_2, \{[93, 94]\}), (p_4, \{[92, 94]\})\}$$

由 Step4 得:

$$OPO = \{(\text{mod}, A_2), (\text{prn}, A_2), (\text{del}, A_2)\}$$

以属性 A_1 的值为 100 的元组存取为例, 由 Step5 得:

$$F^{o-o}(A_2) = \{(700, [92, 93]), (800, [95, 96]), (900, [97, 98])\}$$

由 Step6 得:

$$EOPO(\text{mod}, A_2) = \{(\text{mod}, (700, [92, 93])), (\text{mod}, (800, \varnothing)), (\text{mod}, (900, \varnothing))\}.$$

$$EOPO(\text{prn}, A_2) = \{(\text{prn}, (700, [93, 93])), (\text{pm}, (800, \varnothing)), (\text{prn}, (900, \varnothing))\}.$$

$$EOPO(\text{del}, A_2) = \{(\text{del}, (700, [92, 92])), (\text{del}, (800, \varnothing)), (\text{del}, (900, \varnothing))\}.$$

由 Step7 得:

$$ROPO(r^t) = \{(\text{mod}, (700, [92, 93])), (\text{mod}, (800, \varnothing)), (\text{mod}, (900, \varnothing)), (\text{prn}, (700, [93, 93])), (\text{prn}, (800, \varnothing)), (\text{pm}, (900, \varnothing)), (\text{del}, (700, [92, 92])), (\text{del}, (800, \varnothing)), (\text{del}, (900, \varnothing))\}.$$

由 Step8 和 Step9 得:

$$UOP(u_3^t, r_1^t, [92, 92]) = \{(\text{mod}, (700, [92, 93])), (\text{del}, (700, [92, 92]))\}.$$

4　结论

TRBAC 模型以 $RBAC_3$ 模型为基础, 把时间约束嵌入模型基本要素, 通过角色层次结构上的时态多重继承机制, 有效实现了基于角色的时态对象动态存取控制, 并且支持时态客体层次结构, 模型实现简单, 存取控制决策效率高. 针对 TRBAC 模型的多时间粒度存取控制机制是值得进一步研究的课题.

参考文献:

[1] F Ferraiolo, R Sandhu, R Kuhn. Proposed NIST standard for role-based access control[J]. ACM Transaction on Information and System Security, 2001, 4(3): 224-274.

[2] E Bertino, C Bettini. An access control model supporting periodicity constraints and temporal reasoning[J]. ACM Transactions on Database Systems, 1998, 23(3): 231-285.

[3] L GiuRi, P IgLio. A formal model for role-based access control with constrains[A]. In the Proceedings of the 9th IEEE Workshop on Conr puter Security Foundations[C]. USA: IEEE Press, 1996. 136-145.

[4] Trent Jaeger. On the increasing importance of constraints[A]. In the

Proceedings of the 4th ACM Workshop on Role-Based Access Control [C]. USA: ACM Press, 1999. 33-42.

[5] E Bertino, P A Bonatti, E Ferrari. TRBAC: A temporal role-based access control model [J]. ACM Transactions on Information and System Security, 2001, 4(3): 58-90.

[6] S Barker. TRBAC^{N}: A Temporal Authorization Model [M]. Lecture Notes in Computer Science, Berlin: Springer-verlag Press, 2001.

[7] 王小明, 赵宗涛, 马建峰. 一种新的 RBAC 角色协同关系及其 Petri 网模型[J]. 电子学报, 2003, 31(2): 225-227.
WANG Xiao-ming, ZHAO Zong-tao, MA Jian-feng. A novel role coordination relation of RBAC and its Petri net model [J]. Acta Electronica Sinica, 2003, 31(2): 225-227. (Chinese Source)

[8] 赵庆松, 孙玉芳, 张晓平. 基于角色的域-类型增强存取控制模型研究及其实现[J]. 电子学报, 2003, 31(6): 842-846.
ZHAO Qing-song, SUN Yu-fang, ZHANG Xiao-ping. Research and implementation of role-based domain and type enforcement access control model [J]. Acta Electronica Sinica, 2003, 31(6): 842-846. (Chinese Source)

作者简介:

王小明 男, 1964 年出生于甘肃省天水市, 教授, 博士, 主要研究方向是系统安全, 存取控制, 数据库与工作流系统安全.
E-mail: wangxm@snnu.edu.cn.

赵宗涛 男, 1944 年出生于江苏省徐州市, 教授, 博导, 主要研究方向是系统安全, 数据库与知识库.