文章编号:1008-1542(2010)06-0558-06

具有时间约束的动态角色的访问控制策略

向华萍[1],付智辉[2],陈红丽[1]

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

摘　要:基于角色的访问控制(RBAC)的规范性和易管理性使其被业界普遍接受并广泛应用,但因缺乏时间约束和动态调整权限状态能力,不能很好地满足时间敏感活动的安全需求。针对 RBAC 的不足,提出一个具有时间约束的动态角色访问控制模型,详细描述了时间系统、时间约束指派的相关定义和函数,分析了系统中权限的 4 种状态以及权限状态转化问题;为了保证权限状态更新的效率及权限状态的有效性,设计了计算权限状态转化的最短时间的相应算法。

关键词:时间约束;权限状态;最短时间;角色的访问控制

中图分类号:TP309.2　　**文献标志码:**A

Dynamic time-constrained and role-based access control strategy

XIANG Hua-ping[1], FU Zhi-hui[2], CHEN Hong-li[1]

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

Abstract: Because of its specification and easy administration, role-based access control (RBAC) has been generally accepted and widely used, but it can't fully meet the security needs of time-sensitive activities for its lack of time constraints and dynamic adjustment permission states. For RBAC's shortcomings, a dynamic time-constrained and role-based access control model (DTC-RBAC) is presented. In DTC-RBAC model, we formally define the concepts and functions about time system and time constraints assignments. Furthermore, we analyze four states of permission and permission states transformation, and design an algorithm for calculating the shortest-time of permission states transformation to ensure validity of permission states and efficiency of permission states update.

Key words: time constraints; permission states; shortest-time; role-based access control

访问控制技术是信息安全的重要保障措施之一,能保护信息资源不被非法访问。目前研究与应用最广泛的访问控制技术是基于角色的访问控制(RBAC),其核心思想是将访问权限分配给角色,通过为用户分配不同的角色,达到赋予用户不同权限的目的。FERRAIOLO 等人给出了标准 RBAC 模型的形式描述[1],并最终形成了 NIST 标准。但 NIST RBAC 模型并没有引入时间约束,虽然也有部分 RBAC 的扩充模型讨论了时间约束问题[2-5],但对时间及时间约束相关的概念描述不充分,也没有考虑到引入了时间约束后权限状态的变换以及权限动态管理。笔者在 NIST RBAC 模型的基础上引入了时间约束,对用户角色指派和角色权限指派增加了时间限制,更好地满足了最小权限原则;还分析了权限状态及其转化问题,并给出了计算引

收稿日期:2010-04-16;责任编辑:陈书欣

基金项目:江西省自然科学基金资助项目(2008GZS0012);江西省科技支撑计划重点资助项目(42108002);华东交通大学校立基金资助项目(09RJ01)

作者简介:向华萍(1976-),女,江西萍乡人,讲师,硕士,主要从事信息安全方面的研究。

起权限状态转化的最短时间的相应算法，提高了系统的安全性和权限管理的效率。

1　NIST RBAC 模型

NIST RBAC 模型由 4 个部件模型组成，分别是 Core RBAC，Hierarchal RBAC 以及 Constraint RBAC 中的 2 个责任分离部件模型[1]。以下是这 4 个部件模型的形式化描述。

定义 1　Core RBAC 定义了构成 RBAC 模型的最小元素集合。

1）U,R,Ops,Obs,S 分别表示所有的用户、角色、操作、对象以及会话的集合；

2）$P=2^{Ops\times Obs}$ 表示所有权限的集合；

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

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

5）assigned_$U:(r:R)\rightarrow 2^U$ 返回指派给角色 r 的用户集：assigned_$U(r)=\{u\in U|(u,r)\in UA\}$；

6）assigned_$P:(r:R)\rightarrow 2^P$ 返回指派给角色 r 的权限集：assigned_$P(r)=\{p\in P|(p,r)\in PA\}$；

7）assigned_$R:(u:U)\rightarrow 2^R$ 返回指派给用户 u 的角色集：assigned_$R(u)=\{r\in R|(u,r)\in UA\}$；

8）session_$R:(s:S)\rightarrow 2^R$ 返回与指定会话相关的角色。

定义 2　Hierarchal RBAC 定义了角色继承关系。$RH\subseteq R\times R$ 是角色集上的偏序关系，记为$\geqslant$，称为角色继承，即

1）$r_1\geqslant r_2\Rightarrow$assigned_$P(r_2)\subseteq$assigned_$P(r_1)\wedge$assigned_$U(r_1)\subseteq$assigned_$U(r_2)$；

2）assigned_$U(r)=\{u\in U|(\exists r'\in R|r'\geqslant r\wedge(u,r')\in UA)\}$；

3）assigned_$P(r)=\{p\in P|(\exists r'\in R|r\geqslant r'\wedge(p,r')\in PA)\}$。

定义 3　Constraint RBAC 在 RBAC 模型中添加了责任分离关系，包括静态责任分离（SSD）和动态责任分离（DSD）。

1）$SSD\subseteq 2^R\times N$ 是静态责任分离，满足：$\forall(rs,n)\in SSD,\forall t\in rs:|t|\geqslant n\Rightarrow\bigcap_{r\in t}$assigned_$U(r)=\varnothing$；

2）$DSD\subseteq 2^R\times N$ 是动态责任分离，满足：$\forall rs\in 2^R,\forall n\in N,(rs,n)\in DSD\Rightarrow n\geqslant 2\wedge|rs|\geqslant n$，并且 $\forall s\in S,\forall rs\in 2^R,\forall r_\mathrm{sub}\in 2^R,\forall n\in N,(rs,n)\in \mathrm{DSD},r_\mathrm{sub}\subseteq rs,r_\mathrm{sub}\subseteq$session_$R(s)\Rightarrow|r_\mathrm{sub}|<n$。

2　时间系统

可以看出，NIST RBAC 模型没有解决时间约束问题。事实上任何授权都受时间约束，例如在一个组织中，有工作时间的限制，在非工作时间内，将不允许某些操作的发生，或者不允许某些用户登录系统。

2.1　时间的定义

现实世界中的时间是一个连续的量，而计算机中所能表示的数据都是离散的，所以需要将现实世界的时间离散化，以便于形式化的描述各种时间约束。

定义 4　$T=\{t_i|t_i\in I\}$，T 是计算机中所有时间点的集合，I 是整数集，$t_i\in I$ 代表计算机所描述的一个时间点。

T 与现实时间不同构，但它们之间存在对应关系，笔者定义 2 个函数 real_T 和 virtual_T，实现 T 与真实的时间之间的相互映射。为讨论方便，设 RT 是真实时间的集合。

定义 5　real_$T:T\rightarrow RT$，返回一个真实的时间，是一个严格递增函数。满足：

$$\forall t_i\in T,\exists rt\in RT,\text{有 } rt=\mathrm{real_}T(t_i);$$

$$\forall t_i,t_j\in T,t_i<t_j\wedge rt_i=\mathrm{real_}T(t_i)\wedge rt_j=\mathrm{real_}T(t_j)\Rightarrow rt_i<rt_j。$$

定义 6　virtual_$t:RT\rightarrow T$，返回一个 T 中的时间点，是一个多对一的递增函数。满足：

$$\forall rt_i\in RT,\exists t\in T,\text{有 } t=\mathrm{virtual_}T(rt_i);$$

$$\forall rt_i,rt_j\in RT,rt_i<rt_j\wedge t_i=\mathrm{virtual_}T(rt_i)\wedge t_j=\mathrm{virtual_}T(rt_j)\Rightarrow t_i\leqslant t_j。$$

定义 7　$TR=\{(t_s,t_e)|t_s\in T\wedge t_e\in T\wedge t_s\leqslant t_e\}$，$(t_s,t_e)$是由 2 个时间点定义的时间区间。特别地，当 $t_s=t_e$ 时，时间区间就退化为一个时间点，$\varnothing$表示空集。

定义 8 $TRS=\{(t_{s1},t_{e1}),(t_{s2},t_{e2}),\cdots,(t_{sn},t_{en})\}$，是由一系列的时间区间构成的时间区间序列，在一个时间区间序列中，任何 2 个时间区间都没有交集。

在现实世界中，主体的很多操作都是受到周期时间的约束，周期时间区间实际上是一组特殊的有一定规律的时间区间序列。下面给出了周期时间区间的相关定义。

定义 9 $CTR=\{(t_{s0},t_{e0},\mathrm{d}t)\mid t_{s0}\in T\wedge t_{e0}\in T\wedge \mathrm{d}t\in T\wedge t_{s0}\leqslant t_{e0}\}$ 是周期时间，t_{s0} 和 t_{e0} 表示周期时间的一个时间区间的起始时间和结束时间，$\mathrm{d}t$ 表示 2 个区间的间隔时间。即：$CTR=\{\bigcup_{k}(t_{s0}+k\cdot \mathrm{d}t,t_{e0}+k\cdot \mathrm{d}t)\}$。

2.2 时间区间的集合运算

$\cup,\cap,\neg$ 是时间区间上的 3 个运算，对 $\forall tr_1tr_2\in TR$，设 $tr_1=(t_{i1},t_{j1})$，$tr_2=(t_{i2},t_{j2})$，不失一般性，不妨设 $t_{i1}\leqslant t_{i2}$，tr_1 和 tr_2 关系如图 1 所示。

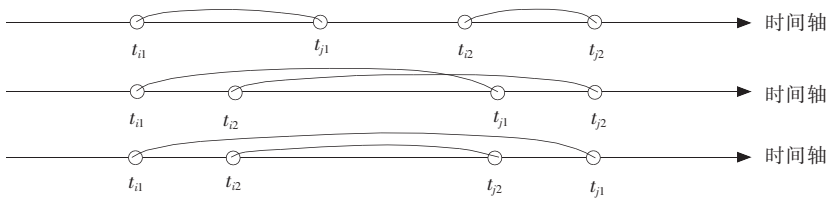

图 1 两个时间区间的关系

Fig. 1 Relation of two time intervals

定义 10 交运算 $\cap$：如果 $t_{j1}<t_{i2}$，则 $tr_1\cap tr_2=\varnothing$；如果 $t_{i2}\leqslant t_{j1}<t_{j2}$，则 $tr_1\cap tr_2=(t_{i2},t_{j1})$；如果 $t_{j1}\geqslant t_{j2}$，则 $tr_1\cap tr_2=(t_{i2},t_{j2})=tr_2$。

定义 11 并运算 $\cup$：如果 $t_{j1}<t_{i2}$，则 $tr_1\cup tr_2=\{tr_1,tr_2\}$，即由 tr_1 和 tr_2 并集构成的时间区间序列；如果 $t_{i2}\leqslant t_{j1}<t_{j2}$，则 $tr_1\cup tr_2=(t_{i1},t_{j2})$；如果 $t_{j1}\geqslant t_{j2}$，则 $tr_1\cup tr_2=(t_{i1},t_{j1})=tr_1$。

定义 12 补运算 $\neg$：$\neg trs_1=\{(t_{-\infty},t_{i1}),(t_{j1},t_{\infty})\}$，$t_{-\infty}$ 和 t_{∞} 分别是时间点的最小值和最大值。

为了统一时间区间的集合运算结果的表示形式，将 $\cap,\cup,\neg$ 3 个运算的结果都转换成时间区间序列。时间区间序列的 $\cap,\cup,\neg$ 运算可以分解成各子时间区间的 $\cap,\cup,\neg$ 运算，鉴于篇幅关系，下面只给出 2 个时间区间序列的 $\cap$ 运算，有关时间区间序列的其他集合运算不在此详述。

定义 13 $trs_1\cap trs_2=\{\bigcup_{i=1,j=1}^{n,m}(tr_{1i}\cap tr_{2j})\}$，

其中 $trs_1,trs_2\in TRS$，且 $trs_1=\{tr_{11},tr_{12},\cdots,tr_{1n}\}$，$trs_2=\{tr_{21},tr_{22},\cdots,tr_{2m}\}$。

2.3 时间的扩展函数

定义 14 $\mathrm{End}T:TR\rightarrow T$ 返回时间区间的结束时间，$\mathrm{End}T(tr)=\{t_e\mid tr=(t_s,t_e)\}$。

定义 15 $\mathrm{is_in}TR:T\times TR\rightarrow\{0,1\}$ 判断时间点 t 是否在时间区间 tr 内，其中 1 为真，0 为假：

$$\mathrm{is_in}TR(t,tr)=\mathrm{is_in}TR(t,(t_s,t_e))=(t\geqslant t_s\wedge t\leqslant t_e)。$$

定义 16 $\mathrm{is_in}TRS:T\times TRS\rightarrow\{0,1\}$ 判断时间点 t 是否在时间区间序列 trs 内：

$$\mathrm{is_in}TRS(t,trs)=\mathrm{is_in}TRS(t,\bigcup_{i}(t_{si},t_{ei}))=\prod_{i}(t\geqslant t_{si}\wedge t\leqslant t_{ei})。$$

定义 17 $\mathrm{is_in}CTR:T\times CTR\rightarrow\{0,1\}$，判断时间点 t 是否在周期时间区间 ctr 内，其中 $ctr=(t_{s0},t_{e0},\mathrm{d}t)$：

$$\mathrm{is_in}CTR(t,ctr)=(t\%\mathrm{d}t\geqslant t_{s0}\%\mathrm{d}t)\wedge(t\%\mathrm{d}t\leqslant t_{e0}\%\mathrm{d}t)。$$

定义 18 $\mathrm{get}TR:TRS\times T\rightarrow 2^{TR}$ 返回时间区间序列 trs 中包含时间点 t 的有效时间区间，如果 trs 的所有时间区间都不包含 t，则返回 $\varnothing$：$\mathrm{get}TR(trs,t)=\{(t_{si},t_{ei})\in trs\mid t\geqslant t_{si}\wedge t\leqslant t_{ei}\}$。

定义 19 $\mathrm{get}TR:CTR\times T\rightarrow 2^{TR}$ 返回周期时间 ctr 中包含时间点 t 的有效时间区间，如果 ctr 的所有时间区间都不包含 t，则返回 $\varnothing$：$\mathrm{get}TR(ctr,t)=\{tr\mid ctr=(t_{s0},t_{e0},\mathrm{d}t)\wedge tr=(t_{s0}+i\cdot \mathrm{d}t,t_{e0}+i\cdot \mathrm{d}t)\wedge(\exists i\in N\mid t\geqslant t_{s0}+i\cdot \mathrm{d}t\wedge t\leqslant t_{ei}+i\cdot \mathrm{d}t)\}$。

3　DTC-RBAC 模型

DTC-RBAC 模型(dynamic time-constrained and role-based access control model)是一个随着时间变化而变化的动态访问控制模型,模型的访问控制实施分为用户角色分配、角色权限分配、动态调整权限状态 3 个部分。

3.1　时间约束的指派

用户的任何访问都受时间约束,本模型通过在进行用户角色指派以及角色权限指派时,增加时间约束,来保证用户访问的时效性。图 2 给出了 DTC-RBAC 模型中基本元素关系。

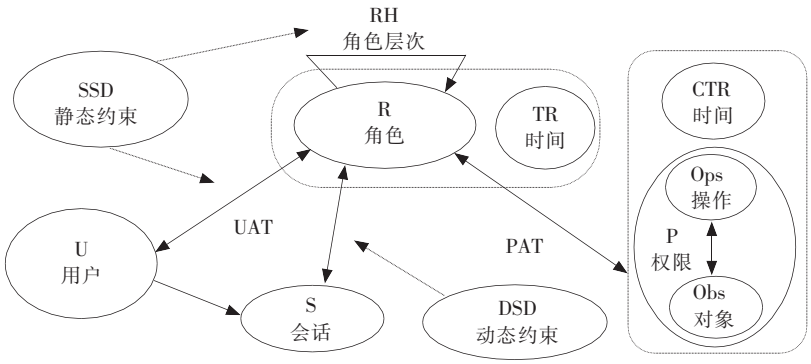

图 2　DTC-RBAC 模型

Fig. 2　DTC-RBAC model

定义 20　$PAT \subseteq P \times R \times CTR$,表示多对多的具有时间约束的角色权限指派。考虑到大多数权限是具有周期性,例如,某省申报科技支撑项目一般是在每年的 9,10 月份,因此角色权限指派的时间约束采用的是周期时间。

定义 21　$UAT \subseteq U \times R \times TR$,表示多对多的具有时间约束的用户到角色的指派关系。用户角色指派的时间约束采用的是时间区间,因为通常都是在一段时间内将一个角色指派给某个用户。

3.2　时间约束的扩展函数

定义 22　$\mathrm{get}P_PAT: R \rightarrow 2^{P \times CTR}$ 返回在角色 r 拥有的所有权限及相应周期时间:

$$\mathrm{get}P_PAT(r) = \{(p, ctr) \mid (p, r, ctr) \in PAT)\}。$$

定义 23　$\mathrm{get}R_UAT: U \rightarrow 2^{R \times TR}$ 返回在角色 r 拥有的所有权限及时间区间:

$$\mathrm{get}R_UAT(u) = \{(r, tr) \mid (u, r, tr) \in UAT)\}。$$

定义 24　$\mathrm{get}P_U: U \rightarrow 2^{P \times TRS}$ 返回在用户 u 拥有的所有权限及相应时间区间序列:

$$\mathrm{get}P_U(u) = \{(p, trs) \mid (\exists (r, tr) \in \mathrm{get}R_UAT(u) \mid (p, r, ctr) \in PAT \wedge trs = tr \cap ctr))\}。$$

定义 25　$\mathrm{valid}P_TR: R \times T \rightarrow 2^{P \times TR}$ 返回角色 r 拥有的所有在时间点 t 有效权限 p 及有效时间区间;如果某个权限的所有时间区间都不包含 t,则是无效权限;如果所有权限都无效,则返回 NULL:

$\mathrm{valid}P_TR(r, t) = \{(p, tr) \mid tr = (t_s, t_e) \wedge (p, ctr) \in \mathrm{get}P_CTR(r) \wedge (t_s, t_e) = \mathrm{get}TR(ctr, t) \wedge (t \geqslant t_s \wedge (t \leqslant t_e))\}$。

定义 26　$\mathrm{invalid}P_TR: R \times T \rightarrow 2^{P \times TR}$ 返回角色 r 拥有的所有在时间点 t 的无效权限 p 及使 p 无效且包含 t 的时间区间:$\mathrm{invalid}P_TR(r, t) = \{(p, tr) \mid (p, r, ctr) \in PAT \wedge \neg \mathrm{is_in}CTR(t, ctr) \wedge ctr = (t_s, t_e, \mathrm{d}t) \wedge i = (t - t_s)/\mathrm{d}t \wedge tr = (t_e + i \cdot \mathrm{d}t, t_s + (i+1) \cdot \mathrm{d}t)\}$ 。

其中:$(t - t_s)/\mathrm{d}t$ 表示 $(t - t_s)$ 整除 $\mathrm{d}t$。

定义 27　$\mathrm{valid}R_TR: U \times T \rightarrow 2^{R \times TR}$ 返回用户 u 拥有的所有在时间点 t 有效角色 r 及有效时间区间;如果某个角色的时间区间都不包含 t,则是无效角色;如果所有角色都无效,则返回 NULL:

$$\mathrm{valid}R_TR(u, t) = \{(r, tr) \mid (u, r, tr) \in UAT \wedge \mathrm{is_in}TR(t, tr))\}。$$

定义 28　$\mathrm{invalid}R_TR: U \times T \rightarrow 2^{R \times TR}$ 返回用户 u 拥有的所有在时间点 t 的无效角色 r 及时间区间:

$$\mathrm{invalid}R_TR(u, t) = \{(r, tr) \mid (u, r, tr) \in UAT \wedge \neg \mathrm{is_in}TR(t, tr))\}。$$

4 动态权限管理

4.1 权限状态

DTC-RBAC模型是一个随着时间、上下文环境变化而变化的动态访问控制模型，其权限管理也是一个动态交互的过程。用户登录时可以选择不同的身份，登录成功后，系统就会为每个用户建立一次会话，并在每次会话中根据用户的身份激活不同的角色，用户通过被激活的角色获得相应权限的集合。由于用户-角色指派(UAT)和角色-权限指派(PAT)具有一定有效期，因此用户的权限也具有时效性，也就是说用户在访问控制过程中，用户权限可能会处于不同的状态。

本模型中，权限状态有如下几种情况。

1)Valid：表示一次会话中在有效期内的权限。

2)Invalid：表示一次会话中不在有效期内的权限。

3)Active：表示一次会话中现在正在执行的，且在有效期内权限的集合。

4)Pause：表示一次会话中由于时间约束或其他原因而无法继续执行的被阻塞的权限的集合。

4.2 与会话相关函数

定义 29 $\text{session_}P: S \rightarrow 2^P$ 返回与指定会话相关的所有权限，即：$\text{session_}P(s)=\{p\in P \mid (p, \text{session_}R(s))\in \pi_{P\times R}PAT\}$。

定义 30 $\text{valid}SR_TR: U\times S\times T\rightarrow 2^{R\times TR}$ 返回一次会话 s 中，用户 u 拥有的所有在时间点 t 有效角色 r 及有效时间区间：$\text{valid}SR_TR(u,s,t)=\{(r,tr) \mid (r,tr)\in \text{valid}R_TR(u,t)\wedge r\in \text{session_}R(s)\}$。

定义 31 $\text{invalid}SR_TR: U\times S\times T\rightarrow 2^{R\times TR}$ 返回一次会话 s 中，用户 u 拥有的所有在时间点 t 的无效角色 r 及时间区间：$\text{invalid}SR_TR(u,s,t)=\{(r,tr) \mid (r,tr)\in \text{invalid}R_TR(u,t)\wedge r\in \text{session_}R(s)\}$。

定义 32 $\text{valid_}SPTR: U\times S\times T\rightarrow 2^{P\times TR}$ 返回一次会话 s 中，用户 u 拥有的所有在时间点 t 有效权限 p 及有效时间区间：$\text{valid}SP_TR(u,s,t)=\{(p,tr) \mid (r,tr_1)\in \text{valid}SR_TR(u,s,t)\wedge (p,tr_2)\in \text{valid}P_TR(r,t)\wedge tr=tr_1\cap tr_2\}$。

定义 33 $\text{invalid}SP: U\times S\times T\rightarrow 2^P$ 返回一次会话 s 中，所有在时间点 t 的无效权限 p：

$$\text{invalid}SP(u,s,t)=\{p \mid p\in \text{session_}P(s)\wedge p\notin \pi_P \text{valid}SR_TR(u,s,t)\}。$$

4.3 权限状态判断时间

每当计算机时间变化时，系统需要自动判断权限状态有效性。这里的关键问题是权限状态判断时间的周期是多长，如果周期时间太长，不能保证权限状态的有效性；如果时间周期太短，系统的运行效率就会降低。为了解决这个问题，本模型是通过先计算引起权限状态变换最短时间 $t_{\min}$，然后在此时间后系统自动调整权限状态。计算 $t_{\min}$ 包括2个过程：计算由有效权限变成无效权限的最短时间 $t_{10\min}$ 和计算由无效权限变成有效权限的最短时间 $t_{01\min}$，则 $t_{\min}=\min(t_{10\min}, t_{01\min})$。

1)计算 $t_{10\min}$ 算法

①将 $t_{1\min}$ 置为时间的最大值；

②获取会话 s 中，用户 u 拥有在当前时间 t_{now} 所有的有效权限 p 及有效时间区间 $(p[i],tr[i])$，$i=1,\cdots,n$，即 $(p[i],tr[i])\in \text{valid}SP_TR(u,s,t_{\text{now}})$；

③for($i=1$;$i<=n$;$i++$)

{$t=\text{End}T(tr[i])$;//获取 tr_i 的结束时间

if(($t-t_{\text{now}}$)$<t_{1\min}$)$t_{1\min}=t-t_{\text{now}}$;}。

2)计算 $t_{01\min}$ 算法

①将 $t_{2\min}$ 置为时间的最大值；

②获取会话 s 中，用户 u 拥有在当前时间 t_{now} 所有无效的角色 r 及时间区间 $(r[i],tr[i])$，$i=1,\cdots,m$，即 $(r[i],tr[i])\in \text{invalid}SR_TR(u,s,t_{\text{now}})$；

③for($i=1$;$i<=m$;$i++$)

{ $t=\text{End}T(tr[i])$

if$((t-t_{now})<t_{2min})t_{2min}=t-t_{now};\}$;

④获取会话 s 中，用户 u 拥有在当前时间 t_{now} 所有有效角色对应的无效权限 p 及时间区间：$(p[i], tr[i]), i=1,\cdots,k$，即 $(p[i],tr[i])\in \text{invalid}P_TR(r,t_{now})\wedge r\in \text{valid}SR_TR(u,s,t_{now})$；

⑤for$(i=1;i<=k;i++)$

$\{t=\text{End}T(tr[i])$

if$((t-t_{now})<t_{2min})t_{2min}=t-t_{now};\}$。

4.4　动态调整权限状态

系统在运行时，根据实际情况自动调整权限状态，主要包括以下几种情况，图 3 显示了权限状态变换过程。

1)用户在登录成功后，系统为每个用户建立一次会话，同时获取用户的有效权限集 Valid 和无效权限集 Invalid，并计算引起权限状态变换最短的时间 t_{min}，则下次变换权限状态的时间 $t=t_{now}-t_{min}$。

2)用户想执行某项权限 p 时，系统先判断 p 是否在 Valid 中，如果在则将 p 添加到 Active 中，否则提示用户权限无效。

3)权限 p 执行结束后或用户取消正在执行的权限，系统从 Valid 中删除权限 p。

4)用户取消被暂停执行的权限，则权限执行终止，系统从 Pause 中删除权限 p。

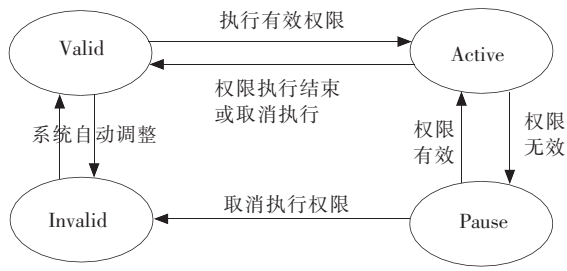

图 3　权限状态迁移图

Fig. 3　Transition of permissions status

5)若到了引起权限状态变换的时间 t，系统将重新获取有效权限集 Valid 和无效权限集 Invalid，并重新计算引起权限状态变换的时间 t；如果在权限状态变换时：①某个在 Invalid 中的权限 p_1 仍在执行，即 $p_1\in$ Active，则暂停执行权限 p_1 并将 p_1 迁移至 Pause；②某个原来被暂停的权限 p_2 现在又变成有效的，则可以继续执行权限 p_2 并将 p_2 迁移至 Active。

5　结　语

从时间约束方面对 NIST RBAC 模型进行了扩展，提出了一个受时间约束的动态角色的访问控制模型，有效地解决了时间敏感活动的访问控制问题，具有更强、更细致的安全描述能力。但由于主要着眼于时间约束及时间相关问题的研究，没有考虑访问控制带来的其他安全因素的影响，因此模型还不是很完善，全面分析访问控制流程相关联问题及其影响，以及时间约束的访问控制在工作流系统中的应用将是下一步的研究方向。

参考文献：

[1]　FERRAIOLO D F, SANDHU R, GAVRILA S, et al. Proposed NIST standard for role-based access control[J]. ACM Transactions on Information and System Security, 2001, 4(3): 224-274.

[2]　MAVRIDIS L, MATTAS A, PAGKALOS I, et al. Supporting dynamic administration of RBAC in web-based collaborative applications during run-time[J]. International Journal of Information and Computer Security, 2009, 2(4): 328-352.

[3]　CHEN L, CRAMPTON J. On spatio-temporal constraints and inheritance in role-based access control[A]. Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security[C]. Tokyo: [s. n.], 2008. 205-216.

[4]　道　炜，汤　庸，冀高峰，等. 基于时限的角色访问控制委托模型[J]. 计算机科学(Computer Science)，2008，35(3)：277-279.

[5]　鲁　柯，周保群，王惠芳. 基于带时间特性 RBAC 的使用控制模型及其管理[J]. 计算机工程(Computer Engineering)，2008，34(6)：170-172.

[6]　张　坤，周万珍，孙秋红. 协同工作(CSCW)中的访问控制模型[J]. 河北工业科技(Hebei Journal of Industrial Science and Technology)，2009，26(5)：315-318.