Journal of Computer Applications
计算机应用,2014,34(2):428 -432
ISSN 1001-9081
CODEN JYIIDU
2014-02-10
http://www.joca.cn

文章编号:1001-9081(2014)02-0428-05
doi:10.11772/j.issn.1001-9081.2014.02.0428

基于属性的用户-角色委派模型可达性分析

任志宇[1,2,3*],陈性元[1,2]
(1.信息工程大学,郑州 450001; 2.河南省信息安全重点实验室,郑州 450004;
3.数学工程与先进计算国家重点实验室,郑州 450001)
(*通信作者电子邮箱 zhiyu.ren@163.com)

摘 要:针对传统基于角色的访问控制(RBAC)管理模型难以表达多样化策略的问题,提出了基于属性的用户-角色委派(ABURA)模型,采用属性作为用户-角色委派的先决条件,丰富了RBAC管理策略的语义。用户-角色可达性分析是验证分布式系统中授权管理策略正确性的重要机制,定义了ABURA模型的用户-角色可达性分析问题,通过分析ABURA模型状态转换特点给出策略约减定理,设计了可达性分析算法,并通过实例对算法进行了验证。

关键词:授权管理模型;可达性分析;属性;角色;用户-角色委派
中图分类号:TP393.08 **文献标志码**:A

Reachability analysis for attribute based user-role assignment model

REN Zhiyu[1,2,3*], CHEN Xingyuan[1,2]
(1. *Information Engineering University, Zhengzhou Henan* 450001, *China*;
2. *Henan Province Key Laboratory of Information Security, Zhengzhou Henan* 450004, *China*;
3. *State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou Henan* 450001, *China*)

Abstract: It is difficult to express diversity policy by traditional RBAC (Role-based Access Control) management model. In order to solve the problem, an Attribute based User-Role assignment (ABURA) model was proposed. Attributes were adopted as prerequisite conditions to provide richer semantics for RBAC management policy. In distributed systems, user-role reachability analysis is an important mechanism to verify the correctness of authorization management policy. The definition of user-role reachability analysis problem for ABURA model was given. According to the characteristics of state transition in ABURA model, some reduction theorems for policy were given. Based on these theorems, user-role reachability analysis algorithm was proposed, and the algorithm got verified through examples.

Key words: authorization management model; reachability analysis; attribute; role; user-role assignment

0 引言

在大规模、分布式系统中,用户、角色、资源的数量巨大,高效、准确地管理各类元素及其之间的关系是实施基于角色的访问控制(Role-based Access Control,RBAC)模型面临的最大困难。在此背景下,需要将RBAC模型在分布式环境下进行扩展,基于角色的授权管理模型应运而生。

1996年,Sandhu等提出了ARBAC97[1]模型,第一次将RBAC扩展到分布式环境下。之后,ARBAC99[2]、ARBAC02[3]、SARBAC[4]、UARBAC[5]等模型被陆续推出。以上模型讨论的重点之一是管理范围的划分,通过can_assign和can_revoke策略中的先决条件,将管理权限划分成不同的管理范围,分别交给不同的管理角色,再将管理角色委派给管理员,实现分布式的授权管理。

在以上模型中,管理策略以用户是否是某些角色的成员作为先决条件,难以表达更加丰富的语义。为了增强模型的表达能力,文献[6-9]提出参数化角色的概念,通过为角色增加参数简化授权策略。例如,文献[9]中的策略can_assign(Dean(*school* = engg), Prof(*dept* = cs), Chair(*dept* = cs)),表示"工程学院的院长可将CS系中的教授任命为系主任"。参数化角色的方法仍以角色作为授权管理策略的先决条件,在实用的过程中具有一定的局限性。对于用户在访问过程中有可能发生改变的一些属性,采用参数化角色的方法难以表达。例如,类似"可委派工作3年以上、计算机专业、已通过入职培训的员工担任质检工程师"的策略,就需要引入用户属性,以增强模型的实用性和可扩展性。另外,采用基于属性的授权管理策略还可为自动授权提供支撑,降低人工授权管理的负担,提高授权效率。

在分布式授权管理模式下,授权管理策略数量巨大,管理策略之间可能存在因果关系,不同管理员的管理操作间也可能会相互作用,很难通过人工检测理解策略的执行结果。策略执行效果的非直观性将直接影响授权管理员的授权决策,带来严重的安全隐患。安全分析可以用于检查策略的安全性和一致性,降低安全风险。

最早对访问控制模型进行安全性分析的是Harrison等于1976年提出HRU模型[10],并对该模型的安全分析问题进行了形式化,提出保护系统主要的安全问题是权限泄露问题。2006年,Li等[11]提出RBAC模型的安全性分析问题(Security

收稿日期:2013-08-21;**修回日期**:2013-10-24。
基金项目:国家973计划项目(2011CB311801);河南省科技创新人才计划项目(114200510001)。
作者简介:任志宇(1974-),女,河南汤阴人,博士研究生,主要研究方向:信息安全、访问控制、授权管理; 陈性元(1964-),男,安徽无为人,教授,博士生导师,博士,主要研究方向:信息安全。

Analysis)的概念,并对安全性分析问题进行了分类。其中简单安全(Simple Safety)问题研究较为深入,又被称为可达性问题,主要分析是否存在某一状态使得不可信用户具有对指定资源的访问权限,其否定回答表示系统是安全的。可达性分析可以帮助管理员制定安全的授权策略,是安全性分析问题中长期备受关注的焦点,其他的安全性分析问题,包括可用性、角色包含等问题可采用相似的方式进行分析。在对可达性分析的研究中,用户-角色可达性分析回答了某位管理员是否可以将某角色委派给指定用户的问题,是众多学者研究可达性问题的切入点,其原因有二:首先用户-角色之间的委派关系变化较为频繁,在 RBAC 的授权管理工作中占有较大工作量,针对用户-角色委派策略的安全分析需求较为迫切;其次,角色-权限可达性、用户-权限可达性都可以通过在用户-角色可达性分析方法的基础上进行归约得到相应的方法。

以 ARBAC97 模型中的 can_assign 策略为例,假设角色 r_1 是 r_2 的父角色,r_2 和 r_3 之间具有静态互斥关系,分析策略 can_assign $(ar,\overline{r_2},r_3)$,该策略表示当某用户未被委派角色 r_2 时,可被担任管理角色 ar 的管理员委派角色 r_3,即对于已委派了 r_2 的用户来说,r_3 是不可达的。而在该策略下,委派了 r_1 的用户对于角色 r_3 来说可达,根据角色之间的权限继承关系,显然违反了职责分离原则。在策略较少的情况下,可通过人工检测的方法避免出现不安全的策略,而大量的策略则必须采用高效的可达性分析算法。

文献[12-13]采用智能规划技术进行可达性分析,文献[14-15]采用图灵机将可达性问题转换为图灵机的停机问题,文献[16]通过构建可达性状态图,通过归约算法实现对目标状态的查询,文献[17]采用模型检测的方法。以上文献均以经典 ARBAC 模型作为研究对象,通过预设一些约束条件,从不同角度给出分析的方法,并研究可达性分析的计算复杂性。文献[9]沿用文献[16]的方法,对参数化 RBAC 模型进行可达性分析。基于属性的授权管理策略其先决条件的表达方式与经典 ARBAC 有所不同,因此,以上文献中的方法均不能直接用于对基于属性的授权管理策略进行分析。

本文提出基于属性的用户-角色委派(Attribute Based User-Role Assignment,ABURA)模型及其管理模型,采用属性表达式作为授权管理策略的先决条件,用户的角色成员关系可作为其中一类属性,因此,ARURA 管理模型的策略比 ARBAC 模型更具有一般性。本文将安全分析问题定位在用户-角色的委派关系上,针对用户-角色可达性开展分析,借鉴文献[9,16]中的分析方法,对 ARURA 模型的用户-角色可达性问题进行了定义,在一定的前提条件下给出了可达性分析算法,并通过实例进行了验证。

1　基于属性的用户-角色委派模型

ARURA 模型及其管理模型的形式化定义如下:

定义 1　ABURA 模型包含以下元素:

1)U 是用户集合,R 是角色集合;2)ATT 是用户可拥有的属性名称的集合;3)UA 是用户-角色委派关系,$(u,r) \in UA$ 表示用户 u 是角色 r 的成员;4)$ATTV$ 是属性名称与属性值之间的关系,$(a,v) \in ATTV$ 表示名为 a 的属性其值为 v;5)$UATT$ 是用户与 $ATTV$ 之间的关系,$(u,attv) \in UATT$,其中 $attv = (a,v) \in ATTV$,表示用户 u 具有 a 属性,当前值为 v。

属性是指用户所具有的某些与授权管理策略相关的特性。用户-角色委派关系也可以作为用户的一种属性,即 $UA \in UATT$,该属性的属性值为用户的已有角色集合。

定义 2　属性表达式。设二元算子 θ,属性名 a,属性值 v。称形如 $(a\theta v)$ 的表达式为属性表达式。

如:$age \geqslant 25$ 表示年龄不小于 25 岁,dep = engineering department 表示单位为工程部,其中 age 和 dep 是属性名称,25 和 engineering department 为属性值,$\geqslant$ 和 = 是算子。

定义 3　ABURA 管理模型的管理策略集合 URA 包含以下策略:

1)can_assign $(ar,P,RT,ATTV)$,其中 ar 表示管理角色,P 为先决条件,RT 表示能够被委派的目标角色集合,$ATTV$ 表示 can_assign 策略执行后发生变化的属性及属性值;

2)can_revoke $(ar,RT,ATTV)$,其中 ar 表示管理角色,RT 表示可被撤销的目标角色集合,$ATTV$ 表示 can_revoke 策略执行后发生变化的属性及属性值。

URA_a 表示 URA 中的所有 can_assign 策略,URA_r 表示 URA 中的所有 can_revoke 策略。

虽然目标角色 RT 可被归入 $ATTV$,但为了便于用户-角色可达性分析,仍将 RT 作为策略中单独的一项。

P 是先决条件,由属性表达式的合取式组成,形如 $ae_1 \wedge ae_2 \wedge \cdots \wedge ae_n$,对能够授权的用户范围进行规范。属性表达式具有表达肯定与否定语义的能力,因此不再显式区分肯定先决条件和否定先决条件,这一点与 ARBAC 模型有所不同。例如,$ae_1 \wedge ae_2 \wedge ae_3$ = ($dept$ = engineering department) $\wedge$ ($age \geqslant 25$) $\wedge$ ($hasrole \neq$ QE),表示工程部年满 25 岁,且未被授予质检工程师的用户。用于表达先决条件的属性表达式称为先决表达式。

属性 $attv$ 与属性表达式 ae 之间有如下关系,其中 $attv = (a,v) \in ATTV$,若 $ae = (a'\theta v')$:

①若 $a = a'$,则称 $attv$ 与 ae 相关,在二者相关的情况下,根据 v 的值,$attv$ 有可能满足 ae,记为 $attv \models ae$;也有可能不满足 ae,记为 $attv \not\models ae$。

②若 $a \neq a'$,则称 $attv$ 与 ae 不相关,记为 $attv \models\!\!\approx ae$。

根据属性在策略执行过程中是否发生变化而将属性分为不变属性和可变属性。不变属性是指不随委派动作 assign、revoke 的执行而发生变化的属性,如用户的年龄、地理位置、网段、安全等级等。可变属性则是指随委派动作的执行而发生变化的属性,如用户已拥有的角色集合、角色数量等,是在 can_assign、can_revoke 策略中出现的那部分属性类型。不变属性与可变属性的划分不是固定的,例如,一般情况下用户的单位属性是不变的,但在策略 can_assign(ar, ($dept$ = productiondivision) $\wedge \neg$ ($hasrole$ = QE),PE, ($dept$ = engineering department)) 下,要求为用户授予 PE 角色后,需要将用户的单位属性改为 engineering department。

2　用户-角色可达性

本文所进行的用户-角色可达性分析基于以下假设:

假设1 模型的管理策略采用开放世界假设(Open World Assumption, OWA)。

开放世界假设即没有规定的是未知的。例如策略 can_assign(ar,$ae_1 \wedge ae_2 \wedge ae_3$,$r$,$ae_4$)表示只要用户满足 ae_1、ae_2 和 ae_3,就可以被担任管理角色 ar 的管理员授予角色 r,而该策略以外的其他属性则不需考虑。

假设2 用户、角色的集合是静态的。

假设在进行用户-可达性分析时,用户和角色的集合是静态的,不会在分析的过程中发生变化。

假设3 属性的类型是有限的,属性值的取值范围可以无限,但策略中的属性表达式的可能性是有限的。

例如,用户具有数据型属性 *dl_count*,表示用户已下载的资源数量,该属性的取值范围是无限的,但该属性作为策略的先决条件时,可能的判断结果是有限的。假设将 *dl_count* 划分为3个等级,根据下载量的不同可被授予不同的角色,策略如下:

can_assign(ar,$dl\_count<99$,r_1,$\varnothing$)

can_assign(ar,$dl\_count\geqslant100$,r_2,$\varnothing$)

can_assign(ar,$dl\_count\geqslant1000$,r_3,$\varnothing$)

其可能性是有限的。

假设4 分离管理约束。

分离管理约束是指将角色分为管理角色和常规角色,要求管理角色只出现在 can_assign 和 can_revoke 策略元组的第一部分。在该约束下,ABURA 管理模型策略不考虑用户和管理角色之间的关系。分离管理约束可以简化用户-角色可达性问题,仅关注用户的常规角色成员关系发生变化的管理动作,这也是可达性分析最关心的问题。因此,可以将 can_assign、can_revoke 策略的第一部分省略。

假设5 每一个用户的角色成员完全独立于其他用户的角色成员。

在此假设下,可以在实现用户-角色可达性分析时,只关注与当前查询用户相关的 UA 关系。

假设6 不考虑角色之间的用户继承关系和权限继承关系。

假设角色之间无用户继承关系和权限继承关系,有角色层次的分析可归约为无角色层次的分析,通过对策略和目标进行转换,使继承的作用变为显式[2]。

基于以上假设,给出 ABURA 状态的定义。

定义4 ABURA 状态 $s=(UATT)$,其中 $UATT$ 是用户 u 所具有的所有属性及属性的当前取值。

为了方便进行用户角色可达性分析,将 UA 从 $UATT$ 中抽取出来,作为状态中独立的一项,记为 $s=(UA,UATT')$,UA 是在状态 s 下用户 u 与常规角色之间的所有关联关系,$UATT'$ 是用户 u 所具有的除 UA 外的其他属性及属性的当前取值,即 $UA \cup UATT' = UATT$。

基于假设5,可以只考虑用户 u 的角色委派关系以及属性情况,并且可以省略 UA 和 $UATT$ 关系中的第一部分。

例如,状态 $s=(\{r_1,r_1\},\{(age,25),(dept,$ engineering department$)\})$表示在状态 s 下,用户 u 被委派了角色 r_1,其年龄为25,所在部门为工程部。

下面给出用户-角色可达性查询问题的定义。

定义5 用户-角色可达性问题。用户-角色可达性查询有如下形式:给定一个用户 u,一个目标角色 r_g,一个管理策略集合 URA,一个管理角色集合 A,设初始状态 $s_0=(UATT_0)$,$(u,r_g)\notin s_0$。问从状态 s_0 开始,是否可由管理员在 URA 的管理策略约束下,将状态 s_0 转换到另一个状态 s_g,使得 u 成为 r_g 的成员,即 $(u,r_g)\in s_g=(UATT_g)$。因此,一个用户-角色可达性查询实例 I 可被表达为 (s_0,URA,u,r_g)。

对于实例 I,用户-角色可达性查询的可通过状态图来表达,顶点(状态)是用户 u 的当前属性,边是实现状态转换的策略。

定义6 用户-角色可达性状态图。用户-角色可达性查询的状态图是有向图(V,E),其中 V 是顶点的集合,E 是边的集合,则:

1) $s=(UATT)=(UA,UATT')\in V$;

2) $(s_1,\varphi,s_2)\in E$,其中 $s_1,s_2\in V,\varphi\in URA$。

3 可达性分析算法

针对某一问题实例 I 进行可达性分析时,可达性状态图中的某些状态可根据以下定理约简。

定理1 添加或更新一个与 URA 中所有先决表达式 UP 无关的属性的策略是可以省略的。

其中,$UP=\{ae \mid \forall\varphi\in URA: ae\in pa(\varphi)\}$,$ae$ 为属性表达式,v 为更新的属性值,$pa(\varphi)$ 表示策略 φ 中用于先决条件的属性表达式集合。

定理2 添加或更新一个与 URA 中所有先决表达式 UP 相关但其属性值不满足 UP 的策略是可以省略的。

定理1、2的证明是显而易见的,因为上述策略所允许的添加和更新操作,不会产生任何有效的转换。

定义6 相关属性表达式。定义相关属性表达式为 $RelAE=\{ae \mid (ae,v)\in posPre^* \wedge gpre\in v\}$。

$RelAE$ 中包含了所有与获得目标角色相关的属性表达式,即:满足这些属性表达式将有助于获得目标角色。$posPre(\varphi)$ 表示 can_assign 策略 φ 的先决表达式与更新后的属性值所组成的二元组,$gpre$ 是添加目标角色 r_g 的 can_assign 策略中的先决表达式。

令 $attv(\varphi)$ 表示 φ 更新的属性值,$rt(\varphi)$ 表示 φ 添加的角色。$posPre(\varphi)$ 可形式化表示为 $posPre(\varphi)=\{(ae,v) \mid \exists\varphi\in URA_u: ae\in pa(\varphi) \wedge attv(\varphi)=v\}$,$gpre$ 可形式化表示为 $gpre=\{ae \mid \exists\varphi\in URA_a: r_g\in rt(\varphi) \wedge ae\in pa(\varphi)\}$。

根据 $RelAE$ 的定义,若 $ae\in RelAE$,则存在一个二元组序列(ae,v),(ae_1,v_1),…,(ae_n,v_n),使得 $gpre\in v_n$,且 $v\models ae_1$、$v_1\models ae_2$,…,$v_{n-1}\models ae_n$。

定理3 属性表达式不在 $RelAE$ 中的 can_assign 策略可省略。

证明 反证法。设 can_assign 策略 φ,$pa(\varphi)\cap RelAE=\varnothing$,假设 φ 对达到目标角色有贡献,不可省略,则存在一条策略路径 $\varphi_1\cdots\varphi_m\varphi\varphi_{m+1}\cdots\varphi_n$,使得依次经过这些策略转换后,可获得目标角色 r_g,即 $rt(\varphi_n)=r_g$。在该路径中,前一条策略更新后的属性值应满足后一条策略的属性表达式,即 $attv(\varphi_m)\models pa(\varphi)$,

$attv(\varphi) \models pa(\varphi_{m+1}),\cdots,attv(\varphi_{n-1}) \models pa(\varphi_n)$。转换为二元组序列$(pa(\varphi_1),attv(\varphi_1)),\cdots,(pa(\varphi),attv(\varphi)),(pa(\varphi_{m+1}),attv(\varphi_{m+1})),\cdots,(pa(\varphi_n),attv(\varphi_n))$。根据定义 6,$pa(\varphi) \subseteq RelAE$,与假设冲突。　　证毕。

定理 4　更新后的属性值不满足 *RelAE* 中属性表达式的 can_assign、can_revoke 策略可省略。

证明　若策略 φ 的属性值不满足 *RelAE* 中的属性表达式,即 $\forall ae \in RelAE, attv(\varphi) \not\models ae$。根据定义 6,针对策略 φ 不存在任意二元组序列$(ae_1,v_1),(ae_2,v_2),\cdots,(ae_n,v_n)$,使得 $attv(\varphi) \not\models ae_1, gpre \in v_n$。因此,$\varphi$ 对获得目标角色不能产生贡献,是可以省略的策略。　　证毕。

根据定理 1 ~4,给出用户-角色可达性分析算法。

输入　问题实例 $I = (s_0, URA, u, r_g)$;

输出　从 s_0 开始运用 *URA* 策略,如果 r_g 可达则返回 true,否则返回 false。

```
1)  initv = attv(s0) ;
2)  for all φ in URA_r do
3)    for all r in s0
4)      if ( r ∈ rt(φ) ) then initv = initv ∪ {attv(φ)} ;
5)  for all φ in URA_a do
6)    if ( r_g ∈ rt(φ) ) then gpre = pa(φ) ; RelAE = {pa(φ)} ;
7)  for all φ in URA do
8)    if (attv(φ) ⊨ UP) or (attv(φ) ⊭ UP) then
        URA = URA - {φ} ;
9)  tmpAE = RelAE;
10) while (tmpAE ≠ ∅) do
11)   tmpAE = ∅;
12)   for all φ in URA_a do
13)     if ( attv(φ) |= RelAE ) then
14)       RelAE = RelAE ∪ {pa(φ)} ;
          tmpAE = tmpAE ∪ {pa(φ)} ;
15) for all φ in URA_a do
16)   if ( ae(φ) ∉ RelAE ) then URA = URA - {φ} ;
17) for all φ in URA do
18)   if ( attv(φ) ⊭ RelAE ) then URA = URA - {φ} ;
19) vset = initv;tmpφ = URA_a ;
20) while ((vset ⊭ gpre) ∧ (tmpφ ≠ ∅)) do
21)   for all φ in URA_a do
22)     tmpφ = ∅;
23)     for all v in vset do
24)       if ( v |= ae(φ) then
25)         tmpφ = tmpφ ∪ {φ} ;
26)         vset = vset - {v} + {attv(φ)} ;
27)         for all φ' in URA_r do
28)           if ( rt(φ) = rt(φ') then
              vset = vset + {attv(φ')} ;
29) if (vset |= gpre) then return true;
30) else return false;
```

算法说明如下:

1)~4)行计算初始属性值集合 *initv*,*initv* 不仅包含初始状态状态中的属性值,还包含将初始状态中的角色集合转换后得到的属性值。转换方法是查找可删除初始角色的 can_revoke 策略,将这些策略更新后的属性值添加到初始角色集中。5) ~ 6) 行将目标角色根据 can_assign 策略转换为目标属性表达式 *gpre*。7) ~ 8) 行根据定理 1、2 对 *URA* 进行约减。9) ~ 14) 行计算相关属性表达式 *RelAE*。15) ~ 18) 行根据定理 3、4 对 *URA* 进行约简。19) ~ 30) 行从初始属性值集合开始进行可达性分析,令当前属性值集合 *vset* = *initv*,查找满足 *vset* 的 can_assign 策略,如果没有合适的策略,则问题不可达,返回 false;如果查找到的策略 φ 为 can_assign 策略,则用 $attv(\varphi)$ 更新 *vset* 中的相应属性值,并将 $rt(\varphi)$ 用相应的 can_revoke 策略转换为属性值,写入 *vset* 中。循环执行上述过程,直到 *vset* 中出现满足 *gpre* 的属性值。

上述算法中,1) ~ 4) 行的计算复杂度为 $O(|rt(\varphi)| \times |URA_r|)$,$|rt(\varphi)|$ 为常量;5) ~ 6) 行的计算复杂度为 $O(|URA_a|)$;7) ~ 8) 行的计算复杂度为 $O(|URA|)$;9) ~ 14) 行的计算复杂度为 $O(|URA_a|^2)$;15) ~ 18) 行的计算复杂度为 $O(|URA_a|+|URA|)$;19) ~ 30) 行的计算复杂度为 $O(|URA_a|^2 \times |vset| \times |URA_r|)$,$|vset|$ 的极大值是常数。因此,算法的计算复杂度为 $O(|URA_a|^2 \times |URA_r|)$。

4　实例分析

以某软件公司的授权管理策略为例进行分析,考虑 3 种属性类型:a_1、a_2、a_3,分别表示单位 *dep*、职责 *duty*、是否是程序员 *pro*。属性 a_1 可能的值包括 v_{11}、v_{12}、v_{13}、v_{14},分别表示公司 COM、研发部 RD、项目组 PT、客户部 AD;属性 a_2 可能的值包括 v_{21}、v_{22}、v_{23},分别表示无职责 none、研发 dev、质检 qos;属性 a_3 可能的值包括 v_{31}、v_{32},分别表示是程序员 yes、不是程序员 no。

本例中与 a_1 相关的属性表达式包括 ae_{11}、ae_{12}、ae_{13},分别表示$(dep = \text{COM})$,$(dep = \text{RD})$,$(dep = \text{PT})$。

与 a_2 相关的属性表达式包括 ae_{21}、ae_{22},分别表示$(duty \neq \text{qos})$,$(duty \neq \text{dev})$。

与 a_3 相关的属性表达式包括 ae_{31},表示$(pro = \text{yes})$。

属性值与属性表达式的满足关系包括:$v_{11} \models ae_{11}$、$v_{12} \models ae_{12}$、$v_{13} \models ae_{13}$、$v_{21} \models ae_{21}$、$v_{21} \models ae_{22}$、$v_{22} \models ae_{21}$、$v_{23} \models ae_{22}$、$v_{31} \models ae_{31}$,其余均不满足。

角色包括:r_1 为普通员工、r_2 为客户部员工,r_3 为研发部员工,r_4 为项目组成员,r_5 为开发人员,r_6 为质检人员,r_7 为高级程序员。其中:r_5 与 r_6 互斥;r_7 是 r_5 的上级角色,也与 r_6 互斥。

设初始状态为 $s_0 = (\{r_1, r_6\})$,目标角色 $r_g = r_7$,分析目标角色是否可达。

将初始状态转换为初始属性值集合 $initv = \{v_{13}, v_{21}\}$,将目标角色转换为目标属性表达式集合 $gpre = \{ae_{31}\}$。

根据定理 1、2 删除 $ua(r_2)$。

计算 $RelAE = \{ae_{11}, ae_{12}, ae_{21}, ae_{21}\}$,根据定理 3、4 删除 $ua(r_6)$、$ur(r_1)$、$ur(r_7)$。

约简后,可执行的策略集合为:$ua(r_1)$、$ua(r_3)$、$ua(r_4)$、$ua(r_5)$、$ua(r_7)$、$ur(r_2)$、$ur(r_3)$、$ur(r_4)$、$ur(r_5)$、$ur(r_6)$。

从初始属性值集合 $\{v_{13}, v_{21}\}$ 开始,在可执行策略集合中查找能够满足先决条件的策略 $ua(r_5)$,当前属性值集合更新为 $\{v_{13}, v_{22}, v_{31}\}$,其中 v_{31} 满足目标属性表达式 ae_{31}。因此,问题实例可达。

表 1 ABURA 管理策略

编号	策略	编号	策略
$ua(r_1)$	can_assign($\varnothing$,r_1,(a_1,v_{11}))	$ur(r_1)$	can_revoke(r_1,$\varnothing$)
$ua(r_2)$	can_assign(ae_{11},r_2,(a_1,v_{14}))	$ur(r_2)$	can_revoke(r_2,(a_1,v_{11}))
$ua(r_3)$	can_assign(ae_{11},r_3,(a_1,v_{12}))	$ur(r_3)$	can_revoke(r_3,(a_1,v_{11}))
$ua(r_4)$	can_assign(ae_{12},r_4,(a_2,v_{21}))	$ur(r_4)$	can_revoke(r_4,(a_1,v_{12}))
$ua(r_5)$	can_assign(ae_{21},r_5,{(a_2,v_{22}),(a_3,v_{31})})	$ur(r_5)$	can_revoke(r_5,{(a_1,v_{13}),(a_2,v_{21}),(a_3,v_{32})})
$ua(r_6)$	can_assign(ae_{22},r_6,(a_2,v_{23}))	$ur(r_6)$	can_revoke(r_6,{(a_1,v_{13}),(a_2,v_{21})})
$ua(r_7)$	can_assign(ae_{31},r_7,$\varnothing$)	$ur(r_7)$	can_revoke(r_7,$\varnothing$)

根据可达性分析过程,还原出如图 1 所示的状态图。

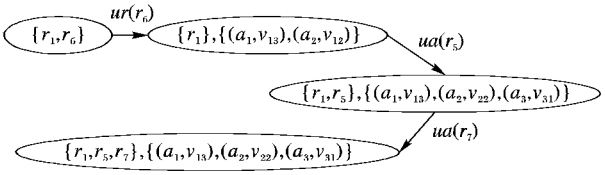

图 1 示例中的可达路径

本例中,只有当删除角色 r_6 之后,才能够被授予 r_7。进一步分析可知,状态({r_5,r_6}) 和 ({r_6,r_7}) 均不可达,表示同一状态下不会出现互斥的角色,表 1 所示的策略支持职责分离原则。

5 结语

本文给出了基于属性的用户-角色委派模型 ABURA,增强了授权管理策略的表达能力,为自动化授权管理提供了可能性。针对该模型的用户-角色可达性问题,给出了可达性分析算法,为保持授权管理策略的正确性和一致性提供支撑。

本文所提出 ABURA 模型中的授权管理策略,其属性的变化是明确的,而实际应用中,有时需要将一些上下文相关因素进行运算之后得出用户当前的属性。这种策略的表达与分析需要得到进一步的研究,使模型的语义更加丰富和实用化。另外,属性表达式之间有可能存在一定的依赖关系,基于这些依赖关系,可以实现策略之间的逻辑推理,提高自动化授权能力。针对这些需求,下一步需要开发更高效的分析算法。

参考文献:

[1] SANDHU R, BHAMIDIPATI V, MUNAWER Q. The ARBAC97 model for role based administration of roles[J]. ACM Transactions on Information and System Security, 1999, 2(1): 105 - 135.

[2] SANDHU R, MUNAWER Q. The ARBAC99 model for administration of roles[C]// Proceedings of the 15th Annual Computer Security Applications Conference. Washington, DC: IEEE Computer Society, 1999: 229 - 238.

[3] OH S, SANDHU R. A model for role administration using organization structure[C]// Proceedings of the 7th ACM Symposium on Access Control Models and Technologies. New York: ACM, 2002: 155 - 162.

[4] CRAMPTON J, LOIZOU G. SARBAC: a new model for role-based administration[R]. London: University of London, 2002.

[5] LI N H, MAO Z Q. Administration in role-based access control [C]// Proceedings of the 2nd ACM Symposium on Information Computer and Communications Security. New York: ACM, 2007: 127 - 138.

[6] LUPU E, SLOMA N M. Reconciling role based management and role based access control[C]// Proceedings of the 2nd ACM Workshop on Role based Access Control. New York: ACM, 1997: 135 - 141.

[7] BACON J, MOODY K, YAO W. A model of OASIS role-based access control and its support for active security[J]. ACM Transactions on Information and System Security, 2002, 5(4): 492 - 540.

[8] GE M, OSBORN S. A design for parameterized roles[C]// Proceedings of the 18th Annual Conference on Data and Applications Security. Berlin: Springer, 2004: 251 - 264.

[9] STOLLER S D, YANG P, GOFMAN M I, *et al.* Symbolic reachability analysis for parameterized administrative role-based access control[J]. Computers & Security, 2011, 30(2/3): 148 - 164.

[10] HARRISON M A, RUZZO W L, ULLMAN J D. Protection in operating systems[J]. Communications of the ACM, 1976, 19(8): 461 - 471.

[11] LI N, TRIPUNITARA M V. Security analysis in role-based access control[J]. ACM Transactions on Information and System Security, 2006, 9(4): 391 - 420.

[12] SASTURKAR A, YANG P, STOLLER S D, *et al.* Policy analysis for administrative role based access control[C]// Proceedings of the 19th IEEE Workshop on Computer Security Foundations. Washington, DC: IEEE Computer Society, 2006: 124 - 138.

[13] LIU Q, JIANG Y, RAO D. Safety analysis of ARBAC policy based on Graphplan[J]. Chinese Journal of Computers, 2009, 32(5): 910 - 921. (刘强, 姜云飞, 饶东宁. 基于 Graphplan 的 ARBAC 策略安全分析方法[J]. 计算机学报, 2009, 32(5): 910 - 921.)

[14] YANG Q, HONG F, YANG M, *et al.* Security analysis on administrative model of role-based access control[J]. Journal of Software, 2006, 17(8): 1804 - 1810. (杨秋伟, 洪帆, 杨木祥, 等. 基于角色访问控制管理模型的安全性分析[J]. 软件学报, 2006, 17(8): 1804 - 1810.)

[15] JHA S, LI N, TRIPUNITARA M, *et al.* Towards formal verification of role-based access control policies[J]. IEEE Transactions on Dependable and Secure Computing, 2008, 5(2): 242 - 55.

[16] STOLLER S D, YANG P, RAMAKRISHNAN C R, *et al.* Efficient policy analysis for administrative role based access control [C]// Proceedings of the 2007 ACM Conference on Computer and Communication Security. New York: ACM, 2007: 445 - 455.

[17] SILVIO R, ANH T, ALESSANDRO A. Boosting model checking to analyse large ARBAC policies[C]// Proceedings of the 8th International Workshop Security and Trust Management. Berlin: Springer, 2012: 273 - 288.