分布式工作流环境下角色匹配的访问控制模型

何思源 欧 博 廖 鑫
(湖南大学信息科学与工程学院 长沙 410082)

摘 要 在分布式工作流环境中,为了使用户获得最合适的权限来执行工作流任务,往往需要给用户指派相应的角色。针对一组给定授权下的用户最佳角色匹配问题,提出一种分布式工作流环境下角色匹配的访问控制模型。该模型可以根据工作流的不同任务,从系统的角色中寻找拥有相关任务执行权限的一组或多组角色集合,然后参考环境、时间约束和角色间的继承关系来进行匹配优化,最终为用户选取最优的角色集合。实验表明,该模型能够剔除冗余角色,为用户精确分配一组最小的角色集合,从而达到角色匹配优化的目的。

关键词 访问控制,分布式工作流,角色匹配,环境和时间约束

中图法分类号 TP309.2 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.07.021

Role Matching Access Control Model for Distributed Workflow

HE Si-yuan OU Bo LIAO Xin
(College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China)

Abstract In the distributed workflow, it is required to assign the users with appropriate roles for the security concerns. This paper proposed a role matching access control model under distributed workflow environment to address the optimal role matching problem for a given authorization. According to different tasks of workflow, the model can find a set or multiple sets of roles with relevant executive authority from the system role, and then optimize the role matching by considering the reference environment, time constraints and the inheritance relationship among the roles. The experimental results show that the model can eliminate redundant roles, and assign a set of minimum set of roles for users, thus achieving the role matching optimization.

Keywords Access control, Distributed workflow, Role matching, Environment and time constraints

1 引言

在分布式环境中,往往需要将某项工作分解成多个相关任务,而这些任务可能分布在不同的应用、进程或者是计算机中,从而处于不同的企业或相同企业的不同部门中,甚至处于不同的网络环境中。完成这项工作中的所有任务所构成的有序业务流程,就形成了工作流。随着互联网科技的快速发展,工作流技术也越来越多地应用在军事、政府、金融、保险等部门,资源信息也由内部网络向资源共享的方向发展。随着用户和角色数量的增多,特别是当资源数据在分布式环境下的工作流中传递时,由此引发的数据泄露等各种安全问题比以前更应得到重视;并且在分布式共享网络环境下,用户还会遭受病毒攻击、黑客攻击以及自然灾害等各种安全问题。因此,在分布式环境中的工作流安全问题需要引起足够的重视。而解决此类安全问题的主要手段之一就是在该环境下引入访问控制技术。所谓访问控制[1-2],就是允许或限制对资源的访问,是针对越权使用资源采取的一种防御措施,它是实现用户数据机密性和进行隐私保护的重要技术手段。对于分布式的工作流环境而言,往往存在对同一客体的访问主体流动性较大的特点,这就需要选择一种合适的、安全的访问控制模型,这也是本文所要研究的问题。

一般来说,访问控制模型[3]是按照特定的访问策略来描述安全系统,并建立安全模型的一种方法。在访问控制研究初期,主要出现了自主访问控制(Discretionary Access Control,DAC)、强制访问控制(Mandatory Access Control,MAC)和基于角色的访问控制(Role Based Access Control,RBAC)等模型。DAC模型[4]就是客体所有者按照自己的安全策略授予系统中的其他用户对客体的访问许可权;这种模型的安全性最低,信息在移动过程中访问权限关系时将被改变,容易造成信息泄露[5]。MAC模型[6]就是系统根据主体和客体的安全属性,以强制的方式控制主体对客体的访问;但其缺点是应用领域比较窄,用户共享数据机制不灵活,在同级别间缺乏

到稿日期:2017-02-21 返修日期:2017-05-20 本文受国家自然科学基金(61502160),教育部博士点新教师基金(20130161120004),中央高校基本科研业务费资助。

何思源(1989-),男,硕士生,主要研究方向为访问控制;欧 博(1985-),男,博士,讲师,硕士生导师,主要研究方向为信息安全,E-mail:oubo@hnu.edu.cn(通信作者);廖 鑫(1985-),男,博士后,讲师,硕士生导师,主要研究方向为信息安全。

控制机制[5]。RBAC 模型[7]允许不同用户以不同角色在相应权限下对客体进行操作，用户通过所指派的角色获得相应的权限，从而实现对资源的访问；但模型中角色本身的管理过于复杂，角色级别越高，授权步骤就越多，这将加大系统管理员的工作量[8]。

随着研究的深入，又先后出现了多种访问控制模型。人们从灵活性、控制粒度、可扩展性等方面对传统模型进行改进，使其向主动型、细粒度、分层次的方向发展，并从任务、属性、行为和信任等新角度来考虑建立模型[3]。王小威等[9]提出的基于任务角色的访问控制(T-RBAC)模型将工作流分解成任务，再将权限通过任务分配给角色，并且权限的分配与任务的上下文有关，还具有明晰的角色层次管理、高扩展性和适应性[10]，从而实现动态分配管理。王静宇等[11]提出的一种面向云计算环境的属性访问控制模型针对目前复杂信息系统的细粒度访问控制和大规模用户动态扩展的问题，通过对主体、客体、环境属性和权限的统一建模，描述授权和访问控制约束，使其具有足够的灵活性和可扩展性。李凤华等[12]提出的基于行为的访问控制安全模型给出了行为的概念及其管理方法，以解决网络环境下支持移动计算的信息系统的访问控制问题。苏铓等[13]将强制访问控制模型与基于行为的访问控制模型相结合，给出实施方案，解决访问控制过程中用户和数据的分级管理等问题。郎波[14]提出的面向分布式系统访问控制的信任度量化模型为用户划分信任级，根据信任的主观性、模糊性与不确定性建立信任度量化模型。付雄等[15]则直接根据用户的实时行为，通过相应算法计算出用户的信任值，并据此为用户分配权限。

为了解决混合角色层次结构中的权限查询、角色激活和角色查找等问题，Joshi 等[16]提出唯一激活集(Uniquely Activable Set，UAS)的概念，方便了角色层次结构的分析，简化了角色查找的过程。对于给定的一组权限，UAS 能在含有角色继承和角色激活关系的混杂角色层次中找出相应的唯一角色集合，但 UAS 集合不能解决查询最优的角色组合问题。杨柳等[17]对 UAS 集合进行优化，提出最小唯一角色集(Minimizing Uniquely Roles，MUR)。MUR 算法根据用户的访问请求来查找角色，在分析不同的访问请求的基础上，得到一组满足用户请求的角色集。该算法在求解效率上相比 UAS 算法有所提高，但是不能满足工作流环境中面向任务请求的安全需求。Zhang 等[18]定义了基于 RBAC 的用户授权查询问题(User Authorization Query problem，UAQ)，其采用贪心算法进行搜索，并利用动态互斥角色约束进行检测，一旦检测到不符合要求的角色组合便停止搜索，权限请求即被拒绝。Lu 等[19]对 UAQ 的不可约性和角色集合权限集的约束这两方面进行了优化，能够有效匹配角色，降低计算复杂性，减少运行时间。

然而以上访问控制模型并不能完全满足分布式工作流环境中对角色的匹配要求。因为在分布式工作流环境中，任务、权限和角色之间的多对多的对应关系更复杂，对于用户的访问控制权限，往往存在多种角色指派方案，这需要系统将许多组不同的角色集合指派给用户来实现同样的授权目的，从而执行系统任务。这种多样化的授权方式会使系统消耗大量的计算资源和存储资源来维持这些指派关系，需要在系统中寻找一组最佳角色集合指派给用户，从而节省系统资源。但是现有 MUR 所确定的最小唯一角色集合并没有考虑到分布式工作流的环境中权限与角色的多对多动态映射关系，使得用户在申请权限时获得的角色集合不一定是唯一的。本文从角色查找这一问题着手，对 MUR 进行改进，提出一种角色匹配的访问控制模型(Role Matching T-RBAC model，RMT-RBAC)。根据分布式环境中不同工作流的任务类型，利用角色查找算法，并增加角色匹配条件，通过不同的环境和时态，依据角色之间继承关系的多少来寻找拥有任务执行权限的一组角色集，最后将该角色集精确匹配给用户。

2 基于角色匹配的访问控制

在分布式工作流环境下，用户数量增多，设置的角色也复杂多变，尤其在交互过程中，可能有多个角色能完成同一保密任务。在为用户组合并匹配角色集的过程中，组合的角色数量为最少的角色集不一定是唯一的。因此，需要增加一些角色匹配条件，然后给用户匹配恰好满足其需求的最佳角色集合。

2.1 提出的访问控制模型

RMT-RBAC 模型主要是在 T-RBAC 模型的基础上进行改进，为分布式工作流环境中执行任务的用户选择匹配的角色。访问控制模型的框架如图 1 所示。

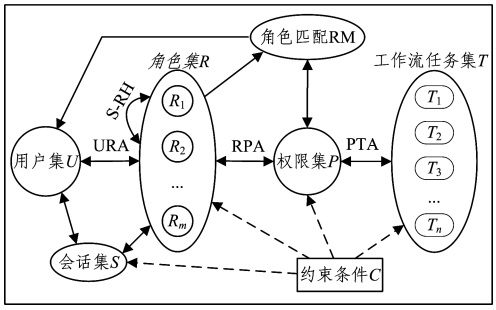

图 1 RMT-RBAC 模型

Fig. 1 RMT-RBAC model

为了执行任务，用户需要向系统申请权限，而用户和角色、角色和权限、权限和任务是多对多的映射关系，系统可以先给用户分配与权限相对应的角色，从而通过角色来执行任务。由于角色和权限是多对多的关系，因此可以通过多个不同的角色来获得同一个权限。若其他用户拥有该权限的其他角色，则可能存在安全隐患。因此，本文在与权限对应的角色集合中进行筛选，引入一个角色匹配机制，以匹配出最佳角色组合来分配给用户，从而提高安全性。

RMT-RBAC 模型的相关概念如下：

用户集 $U=\{U_i \mid i=1,2,\cdots,k\}$：工作流环境中对任务进行操作的主体。

角色集 $R=\{R_i \mid i=1,2,\cdots,m\}$：执行任务的一组集合。$R_i$ 表示 i 中的角色集合，每个 R_i 内又有若干个角色。

权限集 $P=\{P_i \mid i=1,2,\cdots,p\}$：主体能对客体进行的操作。

任务集 $T=\{T_i \mid i=1,2,\cdots,n\}$：用户需要操作的对象。

会话集 S：用户与激活角色之间的映射。当用户激活了部分或全部被授予的角色时，就建立了一个会话。用户执行的权限实际上是在这次会话期间被激活的角色权限。

角色匹配 RM：每需要执行一个任务时，就从相应自治域内的角色集中寻找与该任务匹配的角色集合，并分配权限，同时限制与此集合有继承关系的角色拥有权限。

约束条件 C：对于访问控制中各种指派所做的规则约束。各种指派关系如下：

$URA \subseteq U \times R$，用户到角色的多到多指派关系；

$RPA \subseteq R \times P$，角色到权限的多对多指派关系；

$PTA \subseteq P \times T$，权限到任务的多对多指派关系；

$S\text{-}RH \subseteq R \times R$，角色之间的一些继承关系，有些角色只能部分继承。

2.2　角色匹配

权限和用户相关联，用户通过指派给它的角色来获得权限。角色间存在权限继承关系，使得一种权限可能被多个角色同时拥有，一个角色也有可能拥有多个权限，所以一定的权限可以通过不同的角色组合来获取。因此，在分布式工作流环境中，用户申请一个或多个权限来执行任务时，往往存在很多不同的角色指派方案。

为获取一定权限并保证安全，不能分配冗余权限给用户，需要寻找最佳角色组合并将其匹配给用户。考虑到权限与角色之间是多对多的关系，权限的获取可以通过不同的角色组合获得。图 2 给出了某系统的混合角色层次树，分别用实线、虚线、两端带箭头的实线来表示角色间的继承关系、激活关系、混合关系。若执行某一工作流任务时需要权限 $p=\{p_3, p_4, p_6\}$，则这个权限可以通过以下角色集来获得：$\{\{r_3, r_8\}, \{r_4, r_6\}, \{r_3, r_4\}\}$。由于 r_1 可以继承 r_4 的权限，但它本身就拥有权限 p_1，因此获得的角色集不能有 r_1 这个角色。角色集合 $\{r_2, r_4\}$ 也能获取上述权限，但 r_2 本身拥有权限 p_2，因此该角色集合被舍弃。上面 3 个角色集合满足 MUR 定义，每个角色集合拥有的所有权限（包括角色继承的权限）均不相同，但它们可以通过角色的部分继承获取权限 p，同时也都是最小角色组合。这种权限匹配的最小角色集合不是唯一的，因此在获取多组最小角色集以后，需要一种有效的角色集匹配机制（该机制既要满足最小权限原则，又要保证安全性）来从中匹配合适的角色并指派给用户。

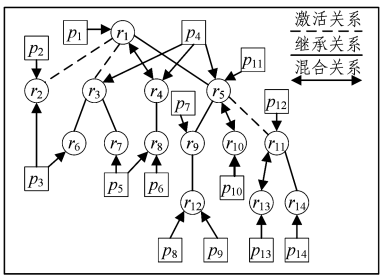

图 2　RMT-RBAC 模型中的层次树

Fig. 2　Level tree of RMT-RBAC model

首先，根据执行任务所处的环境和时态决定选择什么角色。任务所处的环境指执行任务的平台（硬件平台、软件平台等）、位置（场所的物理位置和网络位置等）以及其他与访问控制相关的外部客观信息等，环境的集合用 Env 表示。若角色 r 能在环境 Env 中起作用，则可以表示为 $Env(r) \in Env$；任务所处的时态指在某些时间段需要使用到相关角色来执行任务，时间的集合用 Tim 表示。若角色 r 能在时态 Tim 中起作用，则可以表示为 $Tim(r) \in Tim$。本文使用与安全相关的环境信息和时间信息来约束某些角色分配权限，从而限制拥有这些角色的用户访问任务的资源。

用 R^* 表示通过角色查找算法找到的最小角色集，用 $R^*(i)$ 表示 R^* 中的元素。不同的角色适用于不同的环境和不同的时态。若在环境 Env_m 和时态 Tim_n 情况下，$R^*(i)$ 完全适用环境和时态要求，但 $R^*(j)$ 只适用时态要求，则选用 $R^*(i)$ 角色集。

其次，可以考虑对比角色的继承关系，选择继承关系更少的角色集，以提高系统安全性。当 R^* 中的元素都适用于执行任务所处的环境和时态时，则可以考虑角色继承关系的多少。设 x, y 是角色集 R 中的两个角色，用 $x \geqslant y$ 表示 x 全部继承 y 的权限，用 $x > y$ 表示 x 部分继承 y 的权限，用 $\| x \|$ 表示权限继承的个数。若 R^* 中存在关系 $\| R^*(i) \| \geqslant \| R^*(j) \|$，则选用 $R^*(i)$ 角色集。对于多层次继承，也进行了限制：当不需要中间层次的角色 r 自身拥有的权限，但需要 r 上一层次角色和下一层次角色自身拥有的权限时，r 上一层次的角色不能继承 r 下一层次角色的权限。例如在图 2 中，若需要权限 $\{p_7, p_8, p_{11}\}$，则可以选取 $\{r_5\}$ 作为角色集；若需要权限 $\{p_8, p_{11}\}$，因为中间层次角色 r_9 自身拥有权限 p_7，所以选取 $\{r_5, r_{12}\}$ 作为角色集。通过上述过程，选择出最合适的角色集合匹配给用户。

2.3　角色匹配算法

角色匹配算法的步骤描述如下：

步骤 1　断开角色层次树中的所有激活关系，得到一组独立的只含继承关系的子树；

步骤 2　针对每棵只含继承关系的子树，寻找其中不包含权限继承关系的角色集，即角色集中任意两个角色间都没有权限继承关系；

步骤 3　将所得的不包含权限继承关系的角色集进行任意组合，所有角色组合构成一个集合；

步骤 4　根据完成任务所需要的权限 Per，在上一步所得的集合中选择出最合适的角色组合的集合 $Au(R^*)$；

步骤 5　根据执行任务所处的环境和时态以及角色继承关系的多少，匹配出一个最佳的角色组合 R^* 并将其分配给用户。

本文在 MUR 算法的基础上进行改进和扩充，参照文献[17]中的运算符的定义：$N_1 \otimes N_2 \otimes \cdots \otimes N_m = \{\{x_1 \cup x_2 \cup \cdots \cup x_m\} \mid x_1 \in N_1, x_2 \in N_2, \cdots, x_m \in N_m\}$。假设 $M=\{N_1, N_2, \cdots, N_m\}$，有 $\Theta M = N_1 \otimes N_2 \otimes \cdots \otimes N_m$。

算法描述如算法 1－算法 4 所示。其中，算法 1 是求继承关系的子树；算法 2 是求子树中不含继承关系的角色集；算法 3 是求角色层次 T 的所有 RM 集合；算法 4 是求满足权限的最佳角色集合。

算法 1 求继承关系子树

NodeSet (T)

输入:混合角色层次树 T

输出:角色层次树 T 的所有只含继承关系的子树 I-Her

1. Initialize TempR=∅;//所有被搜寻到的临时角色集合
2. Initialize I-Her=∅;
3. R=TreeNode(T);//R 是角色层次关系 T 中的所有角色集
4. Foreach r∈R Do
5. If r 存在激活关系//如果节点与其双亲节点存在激活关系
6. $I\text{-}Her_i$=BranchTreeNode(T,r);//输出以该节点为根节点的子树
7. R=R−$I\text{-}Her_i$;
8. Return I-Her.

算法 2 求子树中不含继承关系的角色集 Get_RoleSet (T, I-Her)

输入:混合角色层次树 T;只含继承关系的子树 I-Her

输出:只含继承关系的子树中不含继承关系的幂集 I-Her

1. Initialize RoleSet=∅;
2. I-Her=NodeSet(T); //算法 1
3. Foreach IH ∈$2^{I\text{-}Her}$ Do //IH 是 I-Her 的幂集 $2^{I\text{-}her}$中的一个子集
4. If (|IH| == 1) then //如果 IH 中的元素个数为 1
5. RoleSet = RoleSet ∪ IH;//把该元素作为一个子集加入 RoleSet
6. Else if 集合 IH 中的角色间存在继承路径
7. IH = $2^{I\text{-}Her}$−IH;
8. Else RoleSet=RoleSet ∪ IH;
9. Return RoleSet.

算法 3 求角色层次树 T 的所有 RM 集合 RM_ComRoleSet (T,I-Her)

输入:混合角色层次树 T;角色层次树 T 的所有只含继承关系的子树 I-Her

输出:角色层次树 T 中的所有 RM 集合 RM_Set

1. Initialize RM_Set=∅;
2. I-Her=NodeSet(T);//算法 1
3. RoleSet=Get_RoleSet(T,I-Her);//算法 2
4. Foreach RS ∈$2^{RoleSet}$ Do // RS 是 RoleSet 的幂集 $2^{RoleSet}$
5. If (|RS| == 1) then //如果 RS 中的元素个数为 1
6. Foreach r′∈RS Do
7. RM_Set=RM_Set∪{r′};// 把该子集加入 RM_Set
8. Else RM_Set=RM_Set∪ΘRS;
9. Return RM_Set.

算法 4 求满足权限的最佳角色集合 RoleMatching(T,Per)

输入:混合角色层次树 T;执行任务所需要的权限 Per

输出:满足 $Au(R_r^*)$=Per 的最佳集合 R_r^* //$Au(R_r^*)$表示 R_r^* 的权限

1. Initialize Rr^*=∅;
2. Initialize RM=∅;
3. I-Her=NodeSet(T);//算法 1
4. RM_Set=RM_ComRoleSet(T,I-Her);//算法 3
5. Foreach R^* in RM_Set
6. If $Au(R^*)$=Per
7. RM=RM∪R^*;
8. min=‖$R^*(0)$‖;//设置第一个元素集合的继承个数最少
9. Foreach $R^*(i)$ in RM
10. If $Env(R^*(i))\in Env_i$ && $Tim(R^*(i))\in Tim_i$
11. ‖$R^*(i)$‖≤min;
12. R_r^* = R_r^* ∪$R^*(i)$;
13. Return R_r^*.

3 RMT-RBAC 的实现过程

用户在执行任务时,通过角色匹配获取权限的具体过程如图 3 所示。

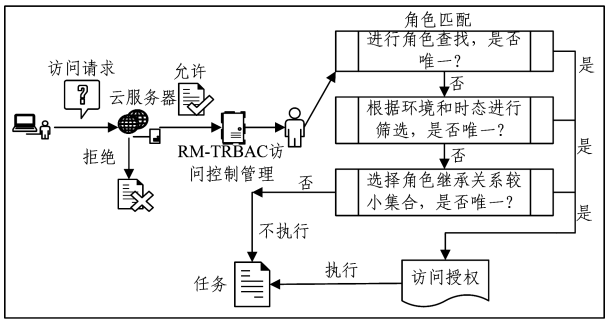

图 3 RMT-RBAC 模型的访问控制实现过程

Fig. 3 Access control process of RMT-RBAC model

步骤 1 用户申请访问。用户向服务器发送访问请求,服务器对用户发送的身份信息进行检查,若身份信息匹配,则允许访问;否则拒绝。

步骤 2 角色匹配。根据执行任务的不同,用户获得不同的权限,RMT-RBAC 访问控制机制为用户匹配相应的角色。首先,根据角色查找算法选择出对应权限的最小角色集合,若该集合是唯一的,则转到步骤 3;否则根据执行任务所处的环境和时态进行筛选,筛选后的角色集合如果是唯一的,则转到步骤 3;否则根据角色集合的继承关系数量进行筛选,转到步骤 3。

步骤 3 访问授权。把筛选后的角色匹配给用户,用户获得角色的相应权限。

步骤 4 任务执行。用户根据获得的权限对任务进行相应的操作。

4 仿真实验

本文对所提出的角色匹配算法进行仿真实验,并将其与 MUR 算法进行比较。以图 2 所示的角色层次树来进行实验对比。实验中环境设为工作环境和公用环境,时态设为上班时间和下班时间,在不同的状态下所选用的角色不同,具体如表 1 所列。

表 1 角色的环境状态、时间状态以及继承关系

Table 1 Role's environmental state, time state and inheritance relationships

时间和环境状态	继承关系
工作、上班	$r_1(13)$,$r_2(0)$,$r_3(2)$,$r_4(1)$,$r_5(7)$,$r_8(0)$,$r_{10}(0)$,$r_{11}(2)$,$r_{12}(0)$,$r_{13}(0)$
工作、下班	$r_2(0)$,$r_3(2)$,$r_7(0)$,$r_8(0)$,$r_9(2)$,$r_{10}(0)$,$r_{11}(2)$
公用、上班	$r_3(2)$,$r_4(1)$,$r_6(0)$,$r_9(2)$,$r_{11}(2)$,$r_{12}(0)$,$r_{13}(0)$
公用、下班	$r_6(0)$,$r_{10}(0)$,$r_{14}(0)$

其中,同一个角色可以在不同的状态下同时使用,例如

r_2 可以在上班状态和下班状态下同时使用，r_{12} 可以在工作状态和公用状态下同时使用。表中角色 r_1(13)表示角色 r_1 能够继承其所有子角色的 13 个权限，以此类推。

对于图 2 所示的角色层次树，本文对比了两种方法，得到的角色组合结果如表 2 所列。需要的权限数为 1～14。针对不同权限数量，随机选取相应个数的权限和不同状态，通过 MUR 算法和角色匹配算法得到不同的角色组合。表 2 中的内容 1－14 分别表示权限 p_1－p_{14}。由表 2 可以看出，MUR 算法查找出的角色集合有一组甚至是多组的情况，这是由于 MUR 能够确定的角色集合相对于用户和角色间的多对多映射而言是唯一的，但在分布式工作流环境中，还需要考虑到权限与角色的动态指派关系，因此用户在申请权限时，通过 MUR 获得的角色集合不一定是唯一的。从表 2 可以看出，角色匹配算法通过增加一些筛选条件后，为用户匹配的角色集合基本上都只有一组，因此得到的角色集合有较好的唯一性，系统的安全性能够得到提升；甚至在公用状态或者是下班状态想获取较多权限时，角色匹配算法不匹配角色供用户使用，避免用户获取权限来进行非法操作，符合现实情况。

表 2　不同状态下申请不同权限的角色组合结果

Table 2　Roles combinations for different states and permissions

权限申请			角色组合序列$\{r_i \mid i=1,2,\cdots,n\}$	
数量	状态	内容	最小唯一角色集合(MUR)	角色匹配(RM)
1	工作、上班	1	{1}	{1}
2	工作、下班	2,3	{2}	{2}
3	工作、上班	3,4,6	{3,8},{4,6},{3,4}	{3,8}
4	工作、上班	2,5,8,13	{2,7,12,13},{2,8,12,13}	{2,8,12,13}
5	工作、下班	6,7,10,12,14	{8,9,10,11}	{8,9,10,11}
6	公用、上班	3,4,6,8,12,13	{2,4,11,12},{3,4,11,12},{3,8,11,12}	{3,4,11,12}
7	工作、下班	2,4,5,7,8,12,14	{2,3,9,11},{2,4,9,11}	{2,3,9,11}
8	公用、下班	1－4,6,8－10	{1,2,4,10,12}	无
9	工作、下班	2－4,6,7,9－11,13	{2,4,5,13}	无
10	公用、上班	1,2,5－7,10－14	{1,2,5,8,11}	无
11	工作、上班	1,3－6,8－10,12－14	{1,3,4,10,11,12},{1,3,8,10,11,12},{1,4,6,10,11,12}	{1,3,4,10,11,12},{1,3,8,10,11,12}
12	公用、下班	2－10,12－14	{2,4,9,10,11}	无
13	工作、上班	2－14	{2,3,4,5,11}	{2,3,4,5,11}
14	工作、上班	1－14	{1,2,3,11}	{1,2,3,11}

结束语　本文提出分布式工作流环境下角色匹配的访问控制模型。针对执行某项任务所需要的一组权限，为了避免角色冗余，在角色集中筛选出角色数量最少的角色集合来获得恰好满足执行任务的权限。但这种角色集合可能有多组，因此我们在筛选过程中加入了环境约束、时间约束以及角色的继承关系，然后从上述多组角色集合中选出一组最佳角色集匹配给用户，通过筛除掉多余的其他组的角色集合，优化了角色匹配问题。未来将会考虑增加角色的筛选条件，对算法进行优化，以降低算法复杂度。

参 考 文 献

[1] WANG Y D, YANG J H, XU C, et al. Survey on Access Control Technologies for Cloud Computing [J]. Journal of Software, 2015, 26(5): 1129-1150. (in Chinese)
王于丁，杨家海，徐聪，等. 云计算访问控制技术研究综述[J]. 软件学报，2015，26(5)：1129-1150.

[2] FENG C S, QIN Z G, YUAN D, et al. Key Techniques of Access Control for Cloud Computing [J]. Acta Electronica Sinica, 2015, 43(2): 312-319. (in Chinese)
冯朝胜，秦志光，袁丁，等. 云计算环境下访问控制关键技术[J]. 电子学报，2015，43(2)：312-319.

[3] LI F H, SU M, SHI Z G, et al. Research Status and Development Trends of Access Control Model [J]. Acta Electronica Sinica, 2012, 40(4): 805-813. (in Chinese)
李凤华，苏铓，史振国，等. 访问控制模型研究进展及发展趋势[J]. 电子学报，2012，40(4)：805-813.

[4] LI N. Discretionary access control[M]// Encyclopedia of Cryptography and Security. Springer US, 2011: 353-356.

[5] HAN D J, GAO J, ZHAI H L, et al. Research Development of Access Control Model [J]. Computer Science, 2010, 37(11): 29-33. (in Chinese)
韩道军，高洁，翟浩良，等. 访问控制模型研究进展[J]. 计算机科学，2010，37(11)：29-33.

[6] UPADHYAYA S. Mandatory Access Control[M]// Encyclopedia of Cryptography and Security. Springer US, 2011: 756-758.

[7] SANDHU R S, COYNE E J, FEINSTEIN H L, et al. Role-based Access Control Models [J]. Computer, 1996, 29(2): 38-47.

[8] ZHANG X M, HUANG Z Q, SUN Y. Research on Privacy Access Control Based on RBAC [J]. Computer Science, 2016, 43(1): 166-171. (in Chinese)
张学明，黄志球，孙艺. 基于 RBAC 的隐私访问控制研究[J]. 计算机科学，2016，43(1)：166-171.

[9] WANG X W, ZHAO Y M. A Task-role-based Access Control Model for Cloud Computing [J]. Computer Engineering, 2012, 38(24): 9-13. (in Chinese)
王小威，赵一鸣. 一种基于任务角色的云计算访问控制模型[J]. 计算机工程，2012，38(24)：9-13.

[10] SEJONG O, SEOG P. Task-role-based Access Control Model [J]. Information System, 2003, 28(6): 533-562.

[11] WANG J Y, FENG L X, ZHENG X F, et al. Research Status and Development Trends of Access Control Model [J]. Journal of Central South University (Science and Technology), 2015, 46(6): 2090-2097. (in Chinese)
王静宇,冯黎晓,郑雪峰. 一种面向云计算环境的属性访问控制模型[J]. 中南大学学报(自然科学版),2015,46(6):2090-2097.

[12] LI F H, WANG W, MA J F, et al. Action-based Access Control Model and Administration of Actions [J]. Acta Electronica Sinica, 2008, 36(10): 1881-1890. (in Chinese)
李凤华,王巍,马建峰,等. 基于行为的访问控制模型及其行为管理[J]. 电子学报,2008,36(10):1881-1890.

[13] SU M, LI F H, SHI G Z. Action-based Multilevel Access Control Model [J]. Journal of Computer Research and Document, 2014, 51(7): 1604-1613. (in Chinese)
苏铓,李凤华,史国振. 基于行为的多级访问控制模型[J]. 计算机研究与发展,2014,51(7):1604-1613.

[14] LANG B. Access Control Oriented Quantified Trust Degree Representation Model for Distributed Systems [J]. Journal on Communications, 2010, 31(12): 45-54. (in Chinese)
郎波. 面向分布式系统访问控制的信任度量化模型[J]. 通信学报,2010,31(12):45-54.

[15] FU X, XU S, ZHOU D M. Research on Trust-based Access Control Model in Cloud Computing Environment [J]. Computer Technology and Development, 2015, 25(9): 139-143. (in Chinese)
付雄,徐松,周代明. 云计算环境下基于信任的访问控制模型研究[J]. 计算机技术与发展,2015,25(9):139-143.

[16] DU S, JOSHI J B D. Supporting Authorization Query and Inter-domain Role Mapping in Presence of Hybrid Role Hierarchy[C]// Proceedings of the 11th ACM Symposium on Access Control Models and Technologies. New York: ACM, 2006: 228-236.

[17] YANG L, TANG Z, LI R F, et al. Roles Query Algorithm in Cloud Computing Environment Based on User Require [J]. Journal on Communications, 2011, 32(7): 169-175. (in Chinese)
杨柳,唐卓,李仁发,等. 云计算环境中基于用户访问需求的角色查找算法[J]. 通信学报,2011,32(7):169-175.

[18] ZHANG Y, JOSHI J B D. Uaq: A Framework for User Authorization Query Processing in RBAC Extended with Hybrid Hierarchy and Constraints[C]// Proceedings of the 13th ACM Symposium on Access Control Models and Technologies. New York: ACM, 2008: 83-92.

[19] LU J, JOSHI J B D, JIN L, et al. Towards Complexity Analysis of User Authorization Query Problem in RBAC [J]. Computers & Security, 2015, 48(C): 116-130.

(上接第 128 页)

[10] KOLDA T G, PROCOPIO M J. Generalized badrank with graduated trust[R]. Sandia National Laboratories, 2009.

[11] LESNIEWSKILAAS C, KAASHOEKM F. Whanau: A sybil-proof distributed hash table[C]// Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation. Berkeley, CA, USA: USENIX Association, 2010: 8.

[12] MOHAISEN A, YUN A, KIM Y. Measuring the mixing time of social graphs[C]// Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. New York, NY, USA: ACM, 2010: 383-389.

[13] BEHRENDS E. Introduction to markov chains: with special emphasis on rapid mixing[M]// Advanced Lectures in Mathematics, 2000.

[14] BLONDEL V D, GUILLAUME J L, LAMBIOTTE R, et al. Fast unfolding of communities in large networks[J]. Journal of Statistical Mechanics: Theory and Experiment, 2008, 2008(10): 155-168.

[15] PANDIT S, CHAU D H, WANG S, et al. Netprobe: A fast and scalable system for fraud detection in online auction networks [C]// Proceedings of the 16th International Conference on World Wide Web. New York, NY, USA: ACM, 2007: 201-210.

[16] SHEBUTI R, LEMAN A. Collective Opinion Spam Detection: Bridging Review Networks and Metadata[C]// Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15). ACM, New York, NY, USA, 2015: 985-994.

[17] PEARL J. Probabilistic reasoning in intelligent systems: Networks of plausible inference[M]. San Francisco: Morgan Kaufmann Publishers Inc., 1988.

[18] JIA J, WANG B, ZHANG L, et al. AttriInfer: Inferring User Attributes in Online Social Networks Using Markov Random Fields[C]// International Conference on World Wide Web. 2017: 1561-1569.

[19] GATTERBAUER W, GUNNEMANN S, KOUTRA D, et al. Linearized and single-pass belief propagation[J]. Proceedings of the Vidb Endowment, 2014, 8(5): 581-592.

[20] WANG B, GONG N Z, FU H. Gang: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs[C]// 2017 IEEE International Conference on Data Mining (ICDM). New Orleans, LA, USA, 2017: 465-474.

[21] SAAD Y. Iterative methods for sparse linear systems (2nd ed) [M]. Reading, MA: Society for Industrial and Applied Mathematics, 2003.

[22] HOLME P, KIM B J. Growing scale-free networks with tunable clustering[J]. Physical Review E, 2002, 65(22): 026107.

[23] BAHNSEN A C, AOUADA D, STOJANOVIC A. Feature engineering strategies for credit card fraud detection[J]. Expert Systems with Applications An International Journal, 2016, 51(C): 134-142.