一种基于TRBAC的动态多级Web服务访问控制模型

陈学龙 郑洪源 丁秋林
(南京航空航天大学计算机科学与技术学院 南京 210016)

摘 要 在分析面向Web服务访问控制研究现状的基础上,针对当前Web服务访问控制模型中存在的不足,提出了一种基于TRBAC的动态多级Web服务访问控制模型(DMWS-TRBAC)。给出了其概念定义、形式化表示及约束规则。新模型引入了服务及其属性,设计了三级访问控制机制,达到了更加安全的细粒度授权。提出了角色扮演者和任务管理者的概念,扩充并严格定义了角色约束和任务约束的内涵,综合考虑时限约束、任务上下文、任务状态及职责分离原则,实现了更加灵活的动态授权。本模型提高了Web服务的安全性,完善了Web服务的访问控制机制,在某军工企业条件保障系统中的初步应用效果良好。

关键词 TRBAC,动态多级,Web服务,访问控制
中图法分类号 TP393 **文献标识码** A

TRBAC-based Dynamic Multilevel Access Control Model for Web Services

CHEN Xue-long ZHENG Hong-yuan DING Qiu-ling
(School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract Based on the research status of access control for Web services and according to the shortcoming in current models, a TRBAC-based dynamic multilevel access control model for Web services was proposed. The model defines some concepts and gives the formal representation and constraint rules. To achieve fine-grained permission, the model introduces services and services attributes, and designs the mechanism of three levels access control. The concepts of Role Actor and Task Manager were presented. The connotation of role constraint and task constraint were extended and defined strictly. The model combines temporal constraint, task context, task state and SSOD principle to describe dynamic permission, and improves the security of Web services and the mechanism of access control for Web services. The preliminary result of the model applied in the condition security system of some defense industry enterprise is good.

Keywords TRBAC, Dynamic multilevel, Web services, Access control

1 引言

访问控制技术[1]是对访问实体进行限制,防止非授权的信息泄露。近年来,随着计算机网络技术的快速发展,企业系统朝着多元化、分布式、异构式的方向发展。Web服务由于其良好的封装性、松散耦合性、灵活性、开放性和可复用性的特点,广泛应用于分布式环境中。然而Web服务是发布在网上的操作接口,采用XML文件和SOAP协议进行信息交互[2],所以与传统的应用相比,Web服务的访问控制面临着诸多的挑战[3]。

目前,在Web Services环境中,主流的访问控制模型有两种:基于属性的访问控制和基于角色的访问控制。基于属性的访问控制模型依据相关实体(包括主体、资源和环境)的属性信息进行授权,虽然能表达动态授权的思想,但由于未引入角色的概念,大量用户的授权管理成为一项极其复杂繁琐的工作,在实现时具有高度的复杂性,且缺少必要的约束条件和继承规则。仅仅基于角色的访问控制模型[4,5]也具有一定的局限性,不能很好地适应Web Services环境的动态特性。文献[6]在TRBAC的基础上引入服务和授权迁移,提出了一个面向工作流和服务的基于角色访问控制模型,但其仅仅是将服务本身进行封装,未从Web服务具体应用的角度对访问资源分而处理,导致粒度较大,在任务执行过程中,动态授权也不灵活。文献[7]虽然提出了面向服务的两级控制(服务级访问控制和属性级访问控制)框架,但没有解决动态授权的问题,且缺少必要的约束规则。针对当前Web服务访问控制模型中存在的粒度较大、动态授权不灵活、约束规则不严格等缺点,本文在TRBAC模型的基础上,提出了一种基于任务和角色的动态多级Web服务访问控制模型(DMWS-TRBAC),该模型引入服务及其属性,设计了三级访问控制机制,将服务及其属性作为资源进行封装并保护起来,对服务本身及其信息的控制更加直接,提出了角色扮演者和任务管理者的概念,综合考虑了角色约束和任务约束,较好地解决了现有模型中存

到稿日期:2013-05-13 返修日期:2013-08-23
陈学龙(1989—),男,硕士生,主要研究方向为知识工程、信息系统与信息安全,E-mail:825174099@qq.com;郑洪源(1973—),男,博士,副教授,硕士生导师,主要研究方向为知识工程、信息系统与信息安全、人机交互;丁秋林(1936—),男,博士,教授,博士生导师,主要研究方向为信息系统、企业信息化。

在的问题，实现了更细粒度、更安全的动态授权。

2 DMWS-TRBAC 模型设计

DMWS-TRBAC 模型与传统面向服务的访问控制模型相比：

(1)将 Web 服务及其属性引入资源范畴，扩充了访问资源的概念，并对访问资源进行划分，引入了功能页面级、服务级和服务属性级等三级访问控制机制。该模型更好地适应了面向服务的计算环境，保护服务本身和服务信息，使服务及其属性的访问更加严格，进一步确保了 Web 服务的安全，完美实现了功能页面级静态保护和服务及属性级动态授权的动静结合的访问控制机制。

(2)提出角色扮演者(Actor)的概念，采用角色的动态管理，使得角色在激活时以及激活后均处于管理和监控的状态，扩充了角色的约束规则，增强了角色的约束检查，杜绝了角色“一次激活，永久使用”的情况，更好地满足了最小特权原则。

(3)提出任务管理者(Task Manager)的概念，通过任务管理者，实现对任务的动态管理，在任务执行过程中，综合考虑任务时限约束、任务上下文、任务状态及职责分离约束，及时动态地调整角色和权限。

角色扮演者和任务管理者的相互配合，使得对 Web 服务的访问权限随着时间、环境和状态信息而变化，权限的授予、持有和收回更加动态、及时，进一步确保了 Web 服务的安全性，提高了 Web 服务访问的灵活性。

DMWS-TRBAC 模型如图 1 所示。

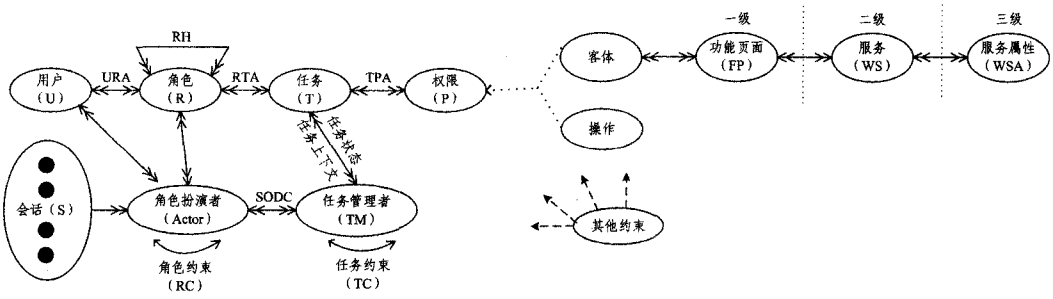

图 1 DMWS-TRBAC 模型图

2.1 模型中组成元素定义

DMWS-TRBAC 模型中除了引用 TRBAC 模型中用户(U)、角色(R)、任务(T)、权限(P)和会话的基本定义外，还扩充了客体的内涵，并引入了一些新的概念，其定义如下：

定义 1 Web 服务(Web Service)。指系统中发布的 Web 服务接口，用以实现具体独立的功能，其集合称为 Web 服务集，记为 WS。

定义 2 Web 服务属性(Web Service Attributes)。指系统中发布的 Web 服务的属性，为服务的调用提供交互信息，它可以是服务的输入参数或返回结果，其集合称为 Web 服务属性集，记为 WSA。服务属性依附于具体的服务，对于某一服务 ws，其属性集记为 $WSA(ws)$，其某一服务属性 $wsa \in WSA(ws)$。

定义 3 客体(Objects)，即访问资源(Access Resource)。指系统中所有可独立标记并能被访问的对象，资源集记为 Resource Set，简记为 RS。

在本模型中，综合考虑传统应用系统和 Web 服务系统的资源特点，可访问的资源包括功能页面(FP)、Web 服务(WS)及 Web 服务属性(WSA)，即 $RS=FP \cup WS \cup WSA$。由此形成了三级访问控制机制：第一级是功能页面级访问控制，对应于传统应用系统中的访问控制机制；第二级是服务级访问控制，第三级是服务属性级访问控制，这两级是根据 Web 服务应用系统的特点而设定的。此三级访问控制间具有良好的拓扑关系：FP 级访问控制是 WS 级访问控制的先决条件，WS 级访问控制又是 WSA 级访问控制的先决条件，所以，只有当角色分配了功能页面的权限时，用户才可能有权限调用服务；角色只有成功通过了服务级访问控制，才可能有权限访问服务属性，从而根据服务属性获取 Web 服务信息进行交互。

将 Web 服务及其属性纳入资源的范畴，改变了以往模型只注重研究基于 XML 文档结构而忽略了保护服务信息的重要性。本模型对服务及其属性的控制更加直接，访问更加严格，适应了 Web 服务应用特点，满足了 Web 服务访问控制需求。

定义 4 角色扮演者(Actor)。角色扮演者是用户在激活角色时，相应产生的一个动态管理角色的对象。角色被激活后，角色扮演者就是该角色的代理，负责角色的所有行为，其集合记为 A，可以用一个三元组〈User, Role, Role Constraint〉表示，其中：

(1)User 是激活角色的用户，即 Actor 所代理的用户，某一用户 $u \in U$；

(2)Role 是用户所激活的角色，即 Actor 所代理的角色，某一角色 $r \in R$；

(3)Role Constraint 是 Actor 所代理的角色的约束，用于角色的约束检查，某一角色约束 $rc \in RC$，这里的角色约束主要包括角色基数约束、角色时限约束、角色前提约束以及角色继承约束。

用户激活一个角色后，系统相应地产生一个 Actor，并将对角色的约束传递到相应的 Actor 中。这样，Actor 将全权代表用户和角色的行为和状态，动态地检查角色的各项约束条件，以满足最小特权和职责分离原则。用户和角色获取服务、执行任务的过程，映射为 Actor 动态获取服务、执行任务的过程。Actor 具有和角色相同的生命周期，当角色失效后，Actor 将被释放，分配给用户的权限将被回收。通过角色扮演者，系统很好地实现了对角色约束的检查以及对用户的角色行为的监控和追踪。

定义 5 任务管理者(Task Manager)。任务管理者是用户在实例化任务时，相应产生的一个动态管理任务的对象，主要负责任务执行期间的约束检查、权限授予、权限调整以及任务非执行期间的权限撤销。任务启动后，任务管理者就是该任务的代理，全程管理任务的执行，其集合记为 TM，可以用

一个三元组〈Role,Task,Task Constraint〉表示,其中:

(1)Role 是启动任务的角色,即任务管理者所代理的角色,某一角色 $r\in R$;

(2)Task 是角色所执行的任务,即任务管理者所代理的任务,某一任务 $t\in T$;

(3)Task Constraint 是任务管理者所代理的任务的约束,用于任务的约束检查,某一任务约束 $tc\in TC$,这里的任务约束主要包括任务基数约束、任务上下文、任务状态以及职责分离约束。

用户在启动任务时,系统相应地产生一个 *TM*,将任务的上下文环境和依赖关系传递到相应的 *TM* 中,并注入其代理角色的 Actor。在任务运行期间,*TM* 管理任务的动作和状态,根据任务时限约束、任务上下文、任务状态以及职责分离约束,及时动态地调整权限,并与其代理的角色的 Actor 配合工作,检查系统其它的约束规则。通过任务管理者,系统权限随任务的执行情况而迁移,很好地实现了权限的动态管理。

2.2 模型形式化表示

DMWS-TRBAC=(U,R,T,P,A,TM,FP,WS,WSA),其中,U 是用户集,R 是角色集,T 是任务集,P 是权限集,A 是角色扮演者集合,TM 是任务管理者集合,FP 是功能页面集合,WS 是 Web 服务集合,WSA 是 Web 服务属性集合。本模型中,除了传统 TRBAC 模型中的 URA、RTA、RH 的关系外,扩充或新定义的关系如下:

(1)任务-权限分配关系(Task Permission Assignment,TPA):$TPA\subseteq T\times P$,即任务与权限之间多对多的映射关系。任务 t 所分配的权限集合 $PA(t)=\{p\in P\wedge(t,p)\in TPA\}$。本模型中,$TPA$ 分为三级:功能页面权限分配、服务权限分配和服务属性权限分配。

(2)功能页面权限分配关系(Function Page Permission Assignment,FPPA):指功能页面权限与任务之间多对多的映射关系,记为 $FPPA$,$FPPA\subseteq TPA$。功能页面权限有允许访问和拒绝访问两种。任务拥有功能页面权限,相应角色的用户才能访问功能页面。

(3)服务权限分配关系(Service Permission Assignment,SPA):指服务权限与任务之间多对多的映射关系,记为 SPA,$SPA\subseteq TPA$。用户在通过功能页面级访问控制后,进入服务级访问控制。服务权限有允许调用和拒绝调用两种。任务拥有服务权限,相应角色的用户才能调用服务。

(4)服务属性权限分配关系(Service Attribute Permission Assignment,SAPA):指服务属性权限与任务之间多对多的映射关系,记为 $SAPA$,$SAPA\subseteq TPA$。用户通过功能页面级访问控制和服务级访问控制后,进入服务属性级访问控制,根据服务属性权限,调用服务并进行信息交互,最终获得 Web 服务信息资源。

(5)功能页面-服务分配关系(Function Page Service Assignment,FPSA):$FPSA\subseteq FP\times WS$,即功能页面与服务间一对多的映射关系。功能页面 fp 所分配的服务集合:$SA(fp)=\{ws\in WS\wedge(fp,ws)\in FPSA\}$。

(6)服务-服务属性分配关系(Service Service-Attribute Assignment,SSAA):$SSAA\subseteq WS\times WSA$,即服务与服务属性之间一对多的映射关系。服务 ws 所分配的服务属性集合 $SSA(ws)=\{wsa\in WSA\wedge(ws,wsa)\in SSAA\}$。

2.3 模型中的约束规则

本模型中的约束规则主要分为角色约束和任务约束,分别由角色扮演者和任务管理者负责检查和管理。为了规范和完善系统的约束规则,进一步保证 Web 服务访问权限动态调整与环境信息的紧密联系,本模型对角色约束和任务约束的内涵进行了扩充和严格定义,下面进行详细阐述。

1. 角色约束

角色约束(Role Constraint,RC),即对角色的约束,规定了任务被赋予角色时,或角色被赋予用户时,以及用户在某一时刻激活一个角色时所应遵循的强制性规则。角色约束主要包括角色基数约束、角色时限约束、角色前提约束以及角色继承约束。

(1)角色基数约束(Role Numerical Constraint,RNC):表示对角色在数值上的约束,包括角色对应的用户成员数目受限,被激活的处于活动状态的角色数目受限,角色对应的任务数目受限等。

(2)角色时限约束(Role Temporal Constraint,RTC):指角色激活在时间上的约束,即角色的生存时间,包括两个时间戳:产生时间和截至时间。$RTC=\{r(t_b,t_e)\}$,表示角色 r 在 (t_b,t_e) 的时间内有效,不在这个时间段,活跃的角色将失效。

(3)角色前提约束(Role Precondition Constraint,RPC):指安全管理员进行授权前当前用户所属角色必须满足的条件,即一个用户被指派到角色 A 的前提是他已经是角色 B,这也反映了角色之间的依赖关系。

(4)角色继承约束(Role Inherit Constraint,RIC):在对角色的公有权限和私有权限划分的基础上,对角色继承在继承方式上进行约束。$RIC=\{r_1,r_2,t,tim\}$,表示角色 r_1 以 tim 方式继承角色 r_2 所执行的任务 t,其中继承方式 tim 分为公有继承、保护继承和私有继承。

2. 任务约束

任务约束(Task Constraint,TC),即对任务的约束,满足相应约束条件的任务才能够得到执行。任务约束主要包括任务基数约束、任务上下文、任务状态以及职责分离约束。

(1)任务基数约束(Task Numerical Constraint,TNC):表示对任务在数值上的约束,主要指任务所分配的角色数目受限。

(2)任务上下文(Task Context,TCT):指的是任务执行所必须满足的工作环境,以保证任务只能在某个特定的时间段、特定的地点以及在任务关系满足的情况下才能被特定的用户执行。如果任务在执行过程中工作环境发生变化导致上下文无法满足,则任务停止执行,相应权限被收回。在本模型中,任务上下文主要包含 3 个方面:

①任务时限约束(Task Temporal Constraint,TTC),即任务活跃的时间范围。$TTC=\{t(t_b,t_e)\}$,表示任务 t 在 (t_b,t_e) 的时间内活跃是有效的,超过这个时间段,活跃状态终止。

②物理环境约束(Physical Environmental Constraint,PEC),主要指任务执行的地点、系统的状态等。对物理环境的要求保证了任务可以顺利、安全地完成。

③任务依赖关系(Task Dependency,TD),指任务之间存在的相互制约关系。任务依赖关系反映了服务的任务特性,包括任务之间和服务本身所具有的依赖关系以及根据安全需要而制定的依赖关系。对于任意两个任务 T_1 和 T_2,可能存

在以下依赖关系：

• 顺序依赖：如果 T_2 必须在 T_1 完成后，才能被激活执行，则存在 $T_1 \rightarrow T_2$；

• 并发依赖：如果 T_1 和 T_2 需同时被激活，则存在 $T_1 \leftrightarrow T_2$；

• 失败依赖：如果 T_2 只能在 T_1 失败后，才会被激活，则存在 $\sim T_1 \rightarrow T_2$；

• 互斥依赖：如果 T_1 和 T_2 不能被同时激活，则存在 $\rightarrow T_1 \rightarrow T_2$ 或 $\rightarrow T_2 \rightarrow T_1$。

(3)任务状态(Task State，TS)：指任务在执行过程中可能经过的状态。令 $t \in T$，则该任务的状态为 $t(state)$。在本模型中，任务有如下几种状态：

• 睡眠状态：表示任务未被激活前的状态。

• 激活状态：任务得到用户请求且运行条件得到满足，则进入该状态。

• 运行状态：任务被激活后开始运行，则进入该状态，访问控制系统授予相关权限，任务活跃期计时开始。

• 无效状态：任务由于某种原因(如出现异常)被终止，进入无效状态，相关权限被收回。

• 挂起状态：任务由于某种原因(如等待资源)被暂停执行，进入该状态，相关权限被冻结，生命活跃期计时仍然继续。

• 完成状态：任务在相关约束下成功结束，则从运行状态进入该状态，相关权限被收回。

在任务执行的过程中，随着任务状态的变化，相关权限也随之而变，从而实现权限的动态管理。任务状态迁移如图 2 所示。

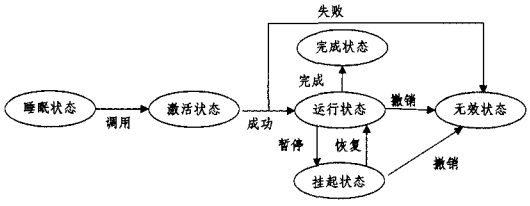

图 2　任务状态迁移图

(4)职责分离约束(Separation Of Duty Constraint，SODC)：又称互斥约束，防止互斥权限被同一实体拥有，以避免欺诈行为。在本模型中，由于任务是角色和权限的桥梁，以任务为单元可以很好地实现职责分离原则。任务互斥关系定义为一些二元组 (t_s, n) 的集合，其中 t_s 是一个任务子集，n 是一个大于 1 的自然数。对于任务互斥关系中的每一个 (t_s, n)，其含义是：任务集合 t_s 中的 n 或 n 以上个任务不能分配给同一角色执行，形式化表示为：

$$\forall t \in t_s : |t_s| \geqslant n \Rightarrow \bigcap_{t \in t_s} assigned\_roles(t) = \emptyset$$

任务互斥约束分为静态任务互斥约束(Static Exclusive Task Constraint，SETC)和动态任务互斥约束(Dynamic Exclusive Task Constraint，DETC)。SETC 要求安全管理员在分配任务时，不允许将互斥任务分配给同一角色或用户。DETC 是指禁止同一角色或用户同时激活并运行两个或多个互斥任务，DETC 是在任务执行过程中进行检查的。

任务互斥约束排除了角色互斥约束的可能性。在任务互斥约束的基础上，角色互斥约束可定义为：对于任意的两个角色 $r_1 \in R$ 和 $r_2 \in R$，$assigned\_tasks(r_1) \cap assigned\_tasks(r_2)$ 存在互斥任务，则角色 r_1 与角色 r_2 互斥，记为 $r_1 | r_2$。

3　访问控制实现

DMWS-TRBAC 模型是根据 Web 服务应用特点，对 Web 服务实施分级保护的动态访问控制模型。该模型在 TRBAC 模型的基础上，引入角色扮演者和任务管理者概念，增强了角色和任务的约束检查，实现了更加严格、灵活的动态授权。模型访问控制实施分为关系分配、角色激活、任务运行(动态权限调整) 3 个阶段。

(1)关系分配。这是系统运行前的准备工作，即安全管理员对用户、角色、任务、权限等关系进行静态分配。在本模型中，URA、RTA 与传统 TRBAC 的对应分配关系一致。由于本模型中引入了服务、服务属性以及三级访问控制机制，任务-权限分配(TPA)稍微复杂，可分为功能页面权限分配(FPPA)、服务权限分配(SPA)、服务属性权限分配(SAPA)。这三级权限分配呈递进关系，环环相扣，访问 Web 服务的过程更加严格，切实保证了 Web 服务的安全性，在实施时，需建立在功能页面-服务分配(FPSA)和服务-服务属性分配(SSAA)的基础上。对于任意的任务 $t \in T$，其获得的相应权限为：

$$TPA(t) = FPPA(t) \cup SPA(t) \cup SAPA(t)$$

其中，$FPPA(t) = \{fp \in FP \wedge (t, fp) \in FPPA\}$，$SPA(t) = \{ws \in WS \wedge (fp, ws) \in FPSA \wedge (t, fp) \in FPPA\}$，$SAPA(t) = \{wsa \in WSA \wedge (ws, wsa) \in SSAA \wedge (fp, ws) \in FPSA \wedge (t, fp) \in FPPA\}$。

对于一些敏感或互斥的任务，需要不同角色或用户执行，在关系分配时，要满足职责分离约束。

(2)角色激活。在第一阶段，虽然安全管理员为用户分配了静态角色，但用户并没有实际权限。用户要获得实际的权限，需要激活相应的角色，即登录系统、创建会话、执行任务。用户在执行任务前，系统会检查用户对应的角色是否具有相应的任务权限，如果有，则创建角色扮演者 Actor，并启动生命期计时器和约束检查机制，如果没有，则直接拒绝相应请求。

Actor 的生成函数用 $Create\_Actor(r)$ 表示，则用户 u 对应的角色 r 在请求任务 t 时，Actor 生成的条件如下：

$$request(u, t) \wedge u \in UA(r) \wedge r \in TA(t) \rightarrow Create\_Actor(r)$$

(3)任务运行，动态权限调整。如果角色扮演者被成功创建且通过预期的角色约束检查，则在任务启动时，系统将创建任务管理者 TM。在任务执行过程中，TM 全程跟踪任务执行情况，及时检查任务约束，综合考虑任务时限、任务上下文、任务状态及职责分离原则，动态地调整任务权限。在服务的进程中，TM 负责任务，Actor 代理角色，二者分工明确，密切配合，充分考虑了服务状态、系统环境和约束规则，动态决定其获取服务的许可，确保了 Web 服务本身以及授权迁移的安全性。

TM 的生成函数用 $Create\_TM(t)$ 表示，则用户 u 对应的角色 r 在请求任务 t 时，TM 生成的条件如下：

$$request(u, t) \wedge u \in UA(r) \wedge r \in TA(t) \wedge Create\_Actor(r) \wedge RC \rightarrow Create\_TM(t)$$

用户通过执行任务最终获得 Web 服务。在执行任务时，只给用户分配所需的权限，未执行任务或任务终止后，用户不再拥有所分配的权限。获得服务 $ws \in WS$ 用函数 $Get\_Service(ws)$ 表示，则用户 u 对应的角色 r 在请求任务 t 时，获得服

(下转第 222 页)

Computers and Operations Research, 1997, 24: 1097-1100

[7] Ganesh K, Narendran T T. CLOVES: A Cluster-and-search Heuristic to Solve the Vehicle Routing Problem with Delivery and Pick-up [J]. European Journal of Operational Research, 2007, 178(3): 699-717

[8] Ostertag L, Doerner K F, Hart R F. A Variable Neighborhood Search Integrated in the POPMUSIC Framework for Solving Large Scale Vehicle Routing Problems[C]// Proceedings of the 5th International Workshop on Hybrid Meta-heuristics, 2008. Málaga, Spain, 2008: 29-42

[9] Vidal T, Crainic T G, Gendreau M, et al. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows[J]. Computers & Operations Research, 2013, 40: 475-489

[10] Högerstrand T. What about people in regional science [C]// Papers and proceedings of the regional science association. 1970, 24: 7-21

[11] Kwan M P, Murray A T, O'Kelly, et al. Recent Advances in Accessibility Research: Representation, Methodology and Applications[J]. Journal of Geographical Systems, 2003, 5: 129-138

[12] Lucasius C B, Dane A D, Kateman G. On K-medoid Clustering of Large Data Sets with the Aid of a Genetic Algorithm: Background, Feasibility and Comparison [J]. Analytica Chimica Acta, 1993, 282(3): 647-669

[13] Solomon M. Algorithms for the vehicle routing and scheduling problems with time window constraints [J]. Operations Research, 1987, 35(2): 254-265

[14] Gehring H, Homberger J. Parallelization of a Two-Phase Metaheuristic for Routing Problems with Time Windows [J]. Journal of Heuristics, 2002, 8(3): 251-276

(上接第 184 页)

务 ws 的条件如下：

$$request(u,t) \wedge u \in UA(r) \wedge r \in TA(t) \wedge (t,fp) \in FPPA \wedge (fp,ws) \in FPSA \wedge (ws,wsa) \in SSAA \rightarrow Get\_Service(ws)$$

4 模型应用与效果分析

目前，DMWS-TRBAC 模型已在某军工企业条件保障系统中得到了初步实现与应用。条件保障系统是一个多节点的分布式系统，通过 Web 服务可以很好地解决异构的问题。军工企业由于其特殊的战略意义，访问控制要求十分严格，且业务流程复杂。在此系统中运用了 DMWS-TRBAC 模型，根据预先定义的权限分配关系、角色约束、任务约束，在业务处理过程中，通过角色扮演者和任务管理者动态地管理权限。实践表明，采用 DMWS-TRBAC 模型减轻了管理员的负担，实现了更加安全灵活的动态授权，从而提高了工作效率。

通过对 DMWS-TRBAC 模型的描述以及在实践中的具体应用可知，新模型从访问资源、角色动态管理、任务动态管理等 3 个方面对已有的访问控制模型做出了改进。新模型具有以下优点：

(1)模型具有通用性和实用性。模型综合考虑了传统访问资源和 Web 服务及其属性，既适用于传统应用系统，也适用于 Web 服务系统，且很好地解决了新兴 Web 服务系统与传统遗留系统整合的问题。在分配关系确定的情况下，模型以任务为中心，沟通角色与权限，访问资源层次清晰，实现很方便。

(2)动态多级访问控制。新模型引入了三级访问控制机制，将功能页面、Web 服务及其属性作为访问资源，对 Web 服务的控制更加直接，访问更加严格，达到了更细粒度的安全授权；提出角色扮演者和任务管理者的概念，在任务执行过程中，综合考虑任务上下文和任务状态，及时动态地调整权限，使得权限与任务的执行情况紧密相连，很好地满足了最小特权原则，具有更好的安全性和灵活性。

(3)更加严格的约束规则。新模型借助角色扮演者和任务管理者，对角色约束和任务约束进行了分类和扩充，并给出了更加完整、严格的定义，实现了强有力的约束规则的检查，支持职责分离原则，权限随着任务执行的情况而变化，实现了权限的动态调整。

结束语 本文针对已有访问控制模型在 Web 服务系统中应用的不足，提出了一种基于任务和角色的动态多级 Web 服务访问控制模型。新模型扩充了访问资源的概念，引入 Web 服务及其属性，并对访问资源进行划分，提出了三级访问控制机制，实现了更细粒度的授权。在动态授权方面，通过角色扮演者和任务管理者来管理和监控任务的执行，综合考虑扩充的角色约束和任务约束，及时动态地调整权限。最后，在某军工企业的条件保障系统中验证了模型的效果。结果表明，本模型的设计，既保证了访问控制的安全性，又具有一定的通用性和实用性，很好地解决了 Web 服务的访问控制问题。

参 考 文 献

[1] Hosseinkhani M, Tarameshloo E, Shajari M. AMVPayword: Secure and efficient anonymous payword-based micropayment scheme[C]// International Conference on Computational Intelligence and Security. 2010: 551-555

[2] Kreger H. Web Services Conceptual Architecture 1. 0[S/OL]. IBM SoftwareGroup. http://www. ibm. com/software/solution/webservices/pdf/WSCA. pdf, 2001

[3] 颜学雄，王清贤，马恒太. Web 服务访问控制模型研究[J]. 计算机科学，2008，35(5)：38-41

[4] 唐金鹏，李玲琳，杨路明. 面向用户属性的 RBAC 模型[J]. 计算机工程与设计，2010，31(10)：2184-2186

[5] 冯翔，甘灵，倪凯，等. 基于 Web Service 的授权访问控制方法[J]. 计算机应用与软件，2007，24(10)：58-59

[6] 许峰，赖海光，黄皓，等. 面向服务的角色访问控制技术研究[J]. 计算机学报，2005，28(4)：686-693

[7] Wonohoesodo R, Tari Z. A role based access control for Web services[C]// IEEE International Conference on Services Computing. Shanghai: IEEE Computer Society Press, 2004: 49-56

[8] 朱一群，李建华，张全海，等. 一种面向 Web 服务的动态分级角色访问控制模型[J]. 上海交通大学学报，2007，41(5)：783-787

[9] 霍远国，马殿富，刘建，等. 面向 Web 服务资源的两层访问控制方法[J]. 计算机科学，2010，37 (7)：125-129

[10] 翟治年，奚建清，卢亚辉，等. 任务状态敏感的访问控制模型及其 CPN 仿真[J]. 西安交通大学学报，2012，46(12)：y1-y7