# 基于 Web 服务的分布式仿真系统的双重访问控制

陈学勤，吴慧中，朱耀琴
（南京理工大学计算机科学与技术学院，南京 210094）

［摘要］ 随着分布式仿真系统与 Web 服务技术的结合日益紧密，以及功能与资源分离的愈加明显，导致大量仿真资源暴露于网络中，传统、单一的访问控制模型已无法应对。为此提出了一种基于属性的针对功能和资源的双重访问控制模型，采用证书代理机制实现功能端单点登录，采用 XACML 实现资源端多属性的访问控制。详细描述了该模型的访问控制流程，目前已应用到虚拟采办系统中。通过对性能的分析和测试，证明了该双重访问控制模型的可用性和有效性。

［关键词］ Web 服务安全；双重访问控制；分布式仿真系统
［中图分类号］ TP393.08 ［文献标识码］ A ［文章编号］ 1009－1742(2009)08－0072－07

## 1 前言

随着以 Web 服务为代表的分布式计算技术不断发展，将仿真与 Web 服务结合起来构建更大规模的分布式仿真系统成为了发展趋势。软件应用程序、硬件抽象或 Web 资源等可以通过 URI 寻址的任意仿真事务被称为仿真资源，并以服务的形式实现仿真资源的单一接口，通过标准的“描述—注册—发现—绑定”协议实现仿真资源的自描述、动态发布、发现、绑定，从而屏蔽平台和语言的差异性，实现跨组织边界的广域网仿真。但同时，采用 Web 服务实现仿真系统使得功能与资源的分离越来越明显，导致大量仿真资源被暴露在广域网中，传统的单一的访问控制模型已经无法应对这种局面，系统的安全性受到严峻考验，故必须解决基于 Web 服务的分布式仿真系统的安全问题。

国际标准组织的文件声明一个安全的信息系统必须具备机密性、完整性和可用性等特征。基于 Web 服务的大规模分布式仿真系统的安全机制，除了需保证上述特性外，还要解决以下几个问题：

1）跨管理域的鉴别与授权，仿真系统的安全机制对于最终用户透明，能够实现用户的单点登陆（single sign on，SSO）；

2）仿真系统的应用功能和资源的双重访问控制，并实现两者授权的一致性；

3）尽量降低安全机制带来的系统开销，减少其对系统性能的影响。

设计一种基于属性的双重访问控制模型，实现仿真功能和资源的双重访问控制以及广域范围内系统的单点登陆、鉴别与授权。采用 XACML（extensible access control markup language）实现资源端的访问控制，并以复杂产品的虚拟采办系统为例进行应用。笔者还对采用 XACML 实现访问控制对系统性能的影响进行了研究，证明了该双重访问控制的可用性和有效性。

## 2 分布式系统的安全基础

信息系统的安全包含多方面的要求，信息储存时要防止非法暴露，信息传递时要防止被窃听和篡改，因此无论是传统的分布式信息系统，还是基于

［收稿日期］ 2007－11－28；修回日期 2008－03－05
［作者简介］ 吴慧中（1942－），女，浙江舟山市人，南京理工大学教授，研究方向为虚拟现实系统仿真；E－mail：wuhzh@njust.edu.cn

Web 服务的松耦合的分布式系统,都需要通过访问控制和保密通信来解决信息的安全问题。

### 2.1 访问控制理论

数学理论支持的加密算法已经发展得比较成熟,因此对于分布式系统而言,安全更重要的是授权、鉴别与访问控制问题。笔者认为访问控制是这三者的核心,鉴别与授权都是为之服务的。访问控制基本形式是一个四元组 $(S,O,A,M)$,其中 $S$ 为主体集合,$O$ 为客体集合,$A$ 为操作权限集合,$M$ 是三元组 $(s,o,a) \in S \times O \times A$ 到 $m \in \{T,F\}$ 的一个函数映射。$M$ 可以表示存储为一个三维的访问矩阵,3 个下标分别对应主体、客体和权限,矩阵的每个值取 $T$ 或 $F$,表示该主体 $s$ 是否拥有对客体 $o$ 进行操作 $a$ 的权限。访问控制的决定过程本质上是对策略库中可适用的策略的评估,实际系统中通常需要进行以下的 3 个步骤:策略管理、身份鉴别和权限检查。

策略管理是对访问控制策略的管理,一般分为两种方式:系统运行过程中由安全管理员手工设定访问控制策略并且在较长时间内保持不变的管理方式,称为静态策略;而系统在运行过程中严密地监视各实体的行为,并根据各实体的当前状态和历史记录动态地调整访问控制策略的方式,称为动态方式。信任代理模型就是一种典型的动态策略。身份鉴别指对某个实体或某些声明的身份的有效性认定,解决“Who are you”的问题。身份鉴别采用的方式主要有 4 种:

1) 本地鉴别;

2) 直接远程鉴别;

3) 第三方参与的远程鉴别;

4) 基于公钥的第三方鉴别。

在开放网络环境中,基于公钥的第三方鉴别因其保密性强、证书便于分发和管理,所以一般使用公钥证书进行身份鉴别。

根据访问主体、客体和授权决策者在授权流程中的分布,访问控制顺序可以分为如下几种[1]:

1) 推顺序;

2) 拉顺序;

3) 代理顺序;

4) 混合顺序。

其差别的实质就是由哪个实体来访问策略决定点,前两者分别是主体和资源,代理顺序中访问策略决定点和充当策略执行点的角色都是 Agent。

### 2.2 Web 服务的安全现状

Web 服务主要包括 3 个层次的安全:点到点的平台/传输层安全、端到端的消息层安全和应用层的安全。平台/传输层安全主要是服务容器自身的安全以及和消息独立的传输协议的安全,一般提供点到点安全,这在现有的 Web 技术中已经较成熟。

Web 服务的 SOAP 消息是基于 XML 语言定义的,消息层安全很大程度上依赖于 XML 文档的安全。W3C 颁布的 XML Encryption[2] 和 XML Signature[3] 标准:前者实现对于 XML 文档的加密,保证其机密性;后者可对 XML 文档的全部或者部分进行签名,保证其数据完整性。这两个标准在 Web 服务安全的研究与实现中都起到了基础作用。W3C 后续又发布了 XKMS (XML key management specification,)[4,5],包含 X - KISS 和 X - KRSS 两部分,定义了 XML 公开密钥的分发和注册协议。各大公司和研究机构也提交了一系列 Web 服务安全的标准草案[6],例如,WS - Security, WS - Policy, WS - Trust, WS - Privacy, WS - Secure Conversation, WS - Federation, WS - Authorization 等。其中 WS - Security 扩展 SOAP 消息,定义了具有大量安全断言及元信息的 SOAP 头,采用 XML Encryption 和 XML Signature 来实现 SOAP 数据的加密和签名。但这些标准(草案)本身并不提供完整的安全解决方案。

OASIS 也批准了 SAML(security assertion markup language)[7] 和 XACML[8] 两个协议,并且不断完善至今。前者是一个基于 XML 的用户鉴别、授权和属性信息交互的框架,主要定义了实体相关的安全断言的文法和处理语义。后者定义了授权策略和授权决定请求与响应的规范模式,还定义了如何评估策略请求并计算得到关于授权决定的响应。这两个协议为实现应用层的安全奠定了基础。

上述是国外厂商和标准组织提出的一系列 Web 服务安全相关的协议和标准,国内很多学者在上述协议和标准的基础上做了不少的应用性的研究工作[9~12]。

## 3 功能和资源的双重访问控制

当前一些实时性要求不高的仿真应用系统越来越多地通过与 Web 服务技术结合以获得屏蔽平台和语言差异性的能力,从而扩大仿真规模和应用范围。而这种技术的结合对仿真系统的安全性提出了新的挑战。为此笔者提出了一种混合顺序的

基于属性的双重访问控制模型,实现对系统功能与资源的双重访问控制;并采用证书代理(proxy)技术实现仿真系统的单点登录和用户的私钥保护。图1为该访问控制模型的原理框图。

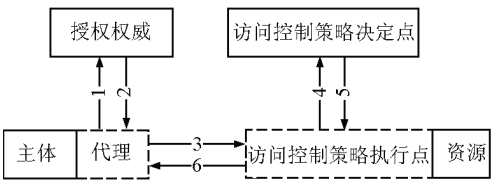


**图1 双重访问控制模型的原理图**

**Fig. 1 Sketch of double access control model**

### 3.1 证书代理

证书代理是声明行为实体可以用代理授予者的身份执行某些操作或发表某些声明,而代理授予者为其正确性负全部责任。本模型在仿真应用程序启动之前系统将利用用户的证书来生成代理,在假设仿真应用程序本身不含有害代码的前提下,用户将为他启动或运行的仿真应用程序所执行的操作负责。这里的证书代理可以认为是公钥证书链的一种临时的延伸。

对于用户 $u$,分别用 $Pk$,$Sk$ 代表 $u$ 的公钥和私钥,其密钥对为 $KP$ 记为 $u.KP = (Pk, Sk)$,证书中心 CA 颁发给 $u$ 的身份证书(Identity Certificate)为

$$Cert(u, Pk, Validity, CA.KP) = ((u \| Pk \| Validity), Sign_{CA.KP}(u \| Pk \| Validity)),$$

则用户 $u$ 授权给代理 $p$ 的代理证书(Proxy Certificate)为

$$Cert(p, p.Pk, Validity, u.KP) = ((p \| p.Pk \| Validity), Sign_{u.KP}(p \| p.Pk \| Validity))。$$

使用证书代理可以给用户私钥足够的保护,私钥在基于 PKI 的安全机制中是至关重要的信息,一般不能在网络中传送,甚至不能够以明文长时间地存储于宿主机中,证书代理可以减少用户私钥的使用次数以降低风险。

使用证书代理的另外一个目的在于能够和访问控制策略决定中心配合实现整个系统的单点登陆,资源端对每次访问请求只需检查代理证书即可完成身份鉴别,不需要用户每次都输入私钥。代理证书也只以内存驻留的方式存在于客户端的系统中,用户退出仿真应用程序时应立即销毁。

### 3.2 双重访问控制

1) 功能的访问控制。功能的访问控制采用推顺序。仿真应用程序充当一个代理角色,用户登录时提交公钥证书,应用程序对用户证书进行鉴别并向主体属性权威(subject attribute authority, SAA)请求主体的属性证书。属性证书的内容是关于访问者的属性,应用程序根据这些属性决定访问控制的结果,决定用户使用具体功能的权限,并生成信任代理,这需要客户端的仿真应用程序具有较强的功能模块配置能力。

2) 资源的访问控制。仿真应用程序端的访问控制已经可以阻止非法授权的用户运行仿真,但是暴露于网络中的仿真资源的安全同样需要考虑。仿真资源端采用拉顺序的访问控制。仿真应用程序向资源发出请求,请求消息附带代理证书以及用户的属性证书。位于资源端接口处的访问控制策略执行点(access control enforcement point, ACEP)接收到请求消息后将首先对代理证书进行鉴别,然后收集当前资源属性以及环境属性,向访问控制决策点(access control decision point, ACDP)发出访问控制的请求,根据 ACDP 响应决定是否返回仿真资源给应用程序。

### 3.3 访问控制流程

对上述模型的原理框图进行细化,设计了一个与实现技术紧密结合的访问控制组件和流程图,如图2所示。图2中给出了安全相关的系统模块,并着重论述双重访问控制模型的流程,对权限分配、访问控制策略优化以及 Web 服务的发布和查找功能将暂不予考虑。

分别用 $U$,$A$,$R$,$E$ 代表用户、仿真应用程序、资源和环境的集合;分别用 $Pk$,$Sk$ 代表实体的公钥和私钥,则密钥对 $KP$ 记为 $KP = (Pk, Sk)$。

证书中心 CA 颁发给用户 $u \in U$ 的身份证书(identity certificate)为

$$Cert(u, u.Pk, Validity, CA.KP) = ((u \| u.Pk \| Validity), Sign_{CA.KP}(u \| u.Pk \| Validity)),$$

其中,

$$Sign_{CA.KP}(u \| u.Pk \| Validity)) = (E_{CA.Sk}(Hash(u \| u.Pk \| Validity)), CA.Pk),$$

以下类似。

主体属性权威 SAA 签署的用户 $u \in U$ 的属性证书(attributes certificate)为

$$Cert(u, u.Attr, Validity, SAA.KP) = ((u \| u.Attr \| Validity), Sign_{SAA.KP}(u \| u.Attr \| Validity)).$$

用户 $u \in U$ 授权给代理 $p$ 的代理证书(proxy certificate)为

$$Cert(p, p.Pk, Validity, u.KP) =$$

$((p||p.Pk||Validity),Sign_{u.KP}(p||p.Pk||Validity))$.

采用该双重访问控制模型的流程描述如下：

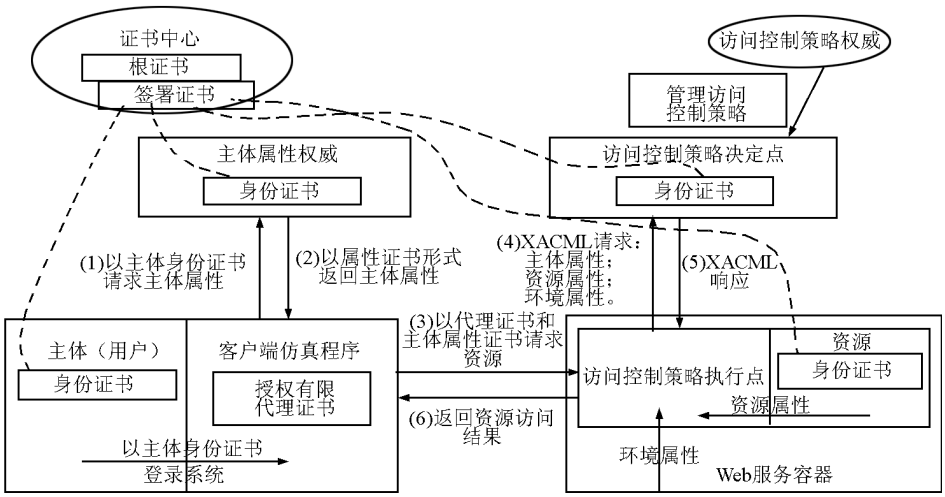


**图 2　安全子系统组件和流程图**

**Fig. 2　Security system components and process**

1）用户 $u \in U$ 提供公钥证书并输入私钥登录仿真应用程序 $a \in A$；

$u \rightarrow a$：　$Cert(u,u.Pk,Validity,CA.KP),u.Sk$

2）仿真应用程序 $a \in A$ 对用户身份进行鉴别后，向 SAA 请求用户属性证书；

$a \rightarrow$SAA：　$(req\_ACMsg,Cert(u,u.Pk,Validity,CA.KP))$

3）SAA 鉴别用户身份后，生成用户 $u$ 的属性证书，并附带自己的身份证书返回 $a$；

SAA$\rightarrow a$：$Cert(u,u.Attr,Validity,PA.KP)$，$Cert(PA,PA.Pk,Validity,CA.KP))$

4）仿真应用程序 $a$ 根据上述用户属性证书决定用户 $u$ 有权使用的功能，并由用户 $u$ 生成代理证书 $Cert(p,p.Pk,Validity,u.KP)$；

5）仿真应用程序 $a$ 向资源提供者 $r \in R$ 发出访问请求消息，附带代理证书、用户属性证书以及用户和 PA 的身份证书；

$a \rightarrow r$：$(req\_Msg,Cert(p,p.Pk,Validity,u.KP),Cert(u,u.Attr,Validity,PA.KP),Cert(u,u.Pk,Validity,CA.KP),Cert(PA,PA.Pk,Validity,CA.KP))$

6）资源接口处的 ACEP 根据请求消息对请求者的身份进行鉴别后，向访问控制 ACDP 发出访问控制请求消息，其中包括资源属性 $r$. Attr 和环境属性 $e$. Attr；

ACEP$\rightarrow$ACDP：$(req\_ACDMsg,Cert(p,p.Pk,Validity,u.KP),r.Attr,e.Attr)$

7）ACDP 向 ACEP 返回访问控制响应消息；

ACDP$\rightarrow$ACEP：　$(rsps\_Msg,Cert(ACEP,ACEP.Pk,Validity,CA.KP))$

8）ACEP 根据 ACDP 的返回结果判断是否允许运行 Web 服务资源，并返回访问结果。

## 4　应用实例

下面以复杂产品的虚拟采办系统为例，说明该双重访问控制模型在实际中的应用。虚拟采办是将仿真技术与复杂产品采办的各个阶段结合起来，通过仿真及其评估技术实现采办的好、快、省的优点，已经建立了计算机支持的虚拟采办的原型系统。在原系统基础上应用上述双重访问控制模型设计并实现了安全子系统，将 Web 服务的消息层安全与应用层安全结合起来，实现功能和资源的双重访问控制，以提升系统的安全性。

原先开发的虚拟采办原型系统采用微软的. Net Framework 2.0 平台开发和运行，安全扩展也将在该平台中进行。. Net 平台已经为 Web 服务的安全规范提供了一些实现，WSE（Web service enhancement）3.0 开发包支持 WS - * 系列规范，为构建安全的 SOAP 消息、实现端到端的消息层安全提供支持。Windows 和 IIS 自身还提供了一整套安全机制，这些都为软件系统的开发提供了良好的基础。此外，XACML. NET 是一个 XACML 在. NET 平台中的开源项目，提供一个 C#的开发包实现了 XACML 协议所定义的模式和基本操作，采用该开发包实现系统

资源端的访问控制。

安全子系统的组件及访问控制流程同图 2 所示,图 3 为资源端 XACML 策略管理中心界面。安全子系统的主要组件包括:

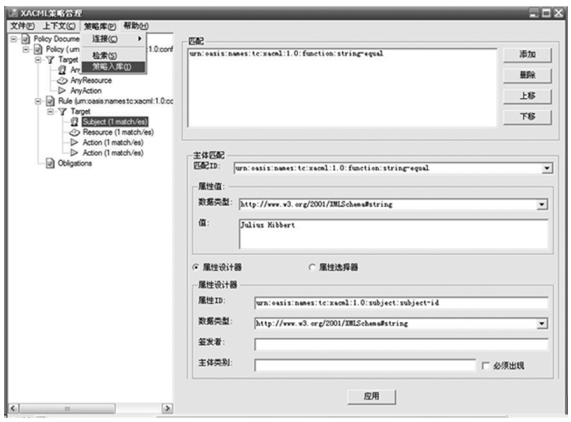

**图 3　资源端 XACML 策略管理**

**Fig. 3　XACML policies management system**

1)证书中心。为各个交互实体签署并管理证书。

2)属性中心。管理用户的属性,鉴别用户身份后将用户属性返回给客户端仿真程序,客户端根据此属性判断用户所能运行的功能。

3)用户代理模块。作为客户端仿真程序的内嵌安全组件,负责应用程序与其他安全组件的交互,并且生成用户代理证书作为请求资源时的凭据。

4)策略管理中心。构建 XACML 访问控制策略,并且对策略库进行管理。

5)访问控制决定模块。根据 XACML 策略和请求作为访问控制决定。

6)访问控制执行模块。作为资源的接口,生成访问控制请求并向 ACDP 请求,根据访问控制结果决定是否响应资源请求。

## 5　性能分析与测试

### 5.1　性能分析

Web 服务系统采用基于 XML 语言的 SOAP 协议来定义消息,XML 是一种跨平台的文本标记语言,其中冗余数据较多,数据表示的有效性较低。而这对于交互频繁的分布式系统来说,其构造和解析 XML 消息的计算时间以及传输 XML 消息的网络延迟将对系统的实时响应有一定影响。

在功能和资源的双重访问控制中,功能的访问控制分为身份鉴别、请求授权以及生成代理 3 个子步骤,只在使用者登录仿真客户端程序的过程中单次执行,对于系统运行后的性能影响不大,暂不予分析。

在引入资源端的访问控制之前,记为 NRAC,客户端程序获取资源所花费的时间主要为消息处理和传输时间,记为 $T(NRAC)$,那么

$$T(NRAC) = T(cstr(req)) + T(trans) + T(destr(req)) + T(proc\_NRAC) + T(cstr(resp)) + T(trans) + T(destr(resp))$$

其中,各时间分别为构建 SOAP 请求、传输 SOAP 请求、解析 SOAP 请求、资源端处理、构建 SOAP 响应、SOAP 响应传输以及解析 SOAP 响应的时间。

采用 XACML 实现访问控制的资源端,记为 RAC,其获取资源花费的时间记为 $T(RAC)$,相对于 $T(NRAC)$ 而言,增量主要在访问控制处理过程中,也即 $T(RAC) = T(NRAC) + T(xac)$,其中,$T(xac)$ 为资源前端的 ACEP 发送请求、得到响应并决定操作的时间,具体为

$$T(xac) = T(cstr(xac\_req)) + T(cstr(xqc\_cnt)) + T(trans) + T(xac\_proc) + T(cstr(xac\_resp)) + T(trans) + T(destr(xac\_resp))$$

其中,各时间分别为构建 XACML 请求、构建 XACML 上下文属性、传输请求、ACDP 处理、构建 XACML 响应、传输响应等时间。

资源端采用访问控制,那么 $T(xac)$ 将是必不可少的,资源端访问控制对系统性能的影响也取决于 $T(xac)$ 在 $T(RAC)$ 中的比例。

### 5.2　性能测试

文章在系统实际运行环境中对 NRAC(见图 4(a))和 RAC(见图 4(b))两种情况下,客户端访问以 Web 服务发布的仿真资源的性能进行了测试和分析。本次测试主要采用 Microsoft 提供的 Visual Studio 2005 Team Edition for Software Testers (VSTE-ST 2005)进行资源访问的负载测试,测试指标为每秒承受的请求数(负载, RpS)以及响应时间(RT),表 1 为系统负载测试环境配置,网络环境为百兆以太网。

首先进行了 10 min 的最大负载测试,采样间隔为 5 s,表 2 为 RAC 和 NRAC 的负载及响应时间测试统计数据,图 5 为 RAC 和 NRAC 的负载及响应时间测试比较曲线。

其次进行了负载与响应时间的关联测试,图 6

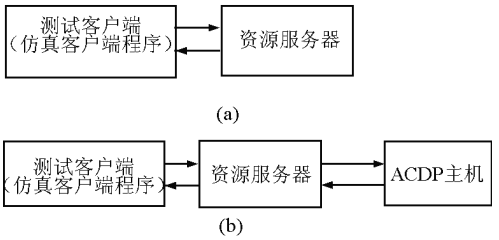


**图 4　NRAC 和 RAC 测试场景**

**Fig. 4　Testing scenes of NRAC and RAC**

为测试启动初期 **0 ~ 60** s 内 RAC 和 NRAC 的响应时间随负载增加的变化曲线。

表 2 的数据和图 5 的曲线显示,在长达 10 min 的测试中,NRAC 的 RpS 值稳定在 220 ~ 230 之间,均值为 225,RAC 的 RpS 均值为 130,NRAC 能承受的最大请求数是 RAC 的 1.73 倍;NRAC 的 RT 均值为 0.034 s,RAC 的 RT 均值为 0.059 s,是 NRAC 的 RT 值的 1.74 倍。RAC 相对于 NRAC 的 RT 增加量主要是上文中 $T(xac)$ 的时间量,在本实例中也即增加了一次 Web 服务访问操作的时间。这对采用 XACML 来进行分布式的访问控制是必需的,由于 RT 值的增加,主机处理性能相同时必将减少 RpS 值。

**表 1　系统负载测试环境配置**

**Tabe 1　Performance analysis and testing environment**

| | Web Service 资源主机配置 | ACDP 主机配置 | 测试工具 |
|---|---|---|---|
| RAC | Intel P4 2.40GHz, DDR 512MB, HD 250GB; Windows Server 2003, .Net Framework v2.0 | Intel PM 1.86Ghz, DDR2 1.5GB, HD 60GB; Windows XP Server, .Net Framework v2.0 | VSTE - ST 2005 |
| NRAC | Intel P4 2.40GHz, DDR 512MB, HD 250GB; Windows Server 2003, .Net Framework v2.0 | Null | VSTE - ST 2005 |

**表 2　RAC 与 NRAC 负载及响应时间测试数据**

**Table 2　Test results of loads and response time**

| | 每秒请求数/$s^{-1}$ | | | 响应时间/s | | |
|---|---|---|---|---|---|---|
| | 最小 | 最大 | 平均 | 最小 | 最大 | 平均 |
| RAC | 75 | 132 | 130 | 0.058 | 0.096 | 0.059 |
| NRAC | 183 | 228 | 225 | 0.034 | 0.064 | 0.034 |

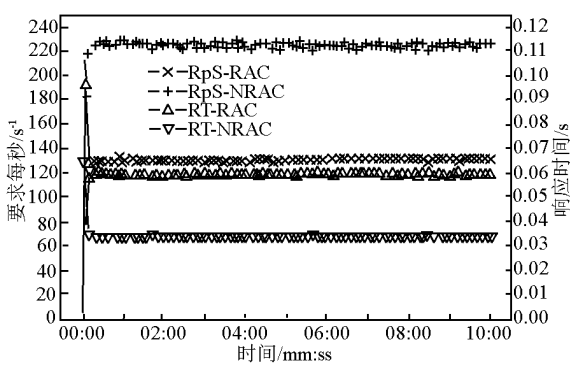


**图 5　RAC 与 NRAC 负载及响应时间测试比较**

**Fig. 5　Test results of loads and response time**

图 6 显示,RAC 场景中,当 RpS < 130 时,RT 值基本上随着 RpS 值线性增加,当 RpS > 130 时,RT 值遽然增大并且波动,RpS 值也无法继续增加,这说明系统已经超越能够持续稳定工作的状态。而 NRAC 场景中,当 RpS < 200 时,RT 值都基本保持平稳,RpS 值继续增加时,RT 值也增大并且波动,系统逐渐稳定于最大负载能力。

性能测试的结果显示,在资源端实施访问控制

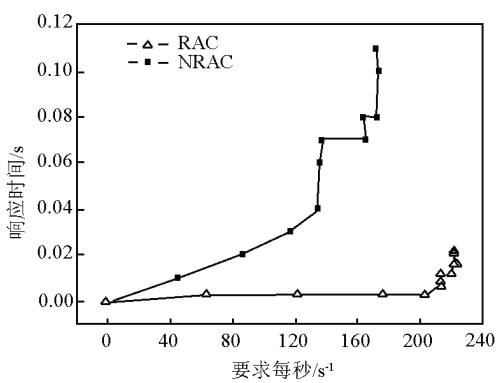


**图 6　RAC 与 NRAC 响应时间随负载变化曲线(0 ~ 60 s)**

**Fig. 6　Change curve of RT with the RpS increasing**

对于系统的负载以及响应时间都有一定的影响,但每秒 130 次请求的处理能力也可以满足准实时的广域网仿真要求。

## 6　结语

针对分布式仿真系统与 Web 服务技术结合面

临安全的挑战,提出并设计了一种混合顺序的基于属性的双重访问控制模型,实现对系统功能与资源的双重访问控制。功能端是一种基本的基于属性的访问控制,通过访问主体属性权威得到访问者的主体属性证书,并据此访问者运行功能的权限。功能端通过受限的证书代理来实现使用者的单点登录,这样还能对访问者的私钥进行最大限度的保护。资源端依赖主体属性、资源属性以及环境属性来决定访问控制。由于访问者主体属性的一致性,其对双重访问控制决定的影响也是一致的。文章以复杂产品的虚拟采办系统为例对双重访问控制模型进行了实现和应用,采用 XACML 实现了资源端的访问控制,建立了 XACML 策略库及管理系统,并且对访问控制的性能进行了分析与测试。性能测试的结果说明该访问控制模型对于性能有一定影响,但仍然可以满足广域网中的分布式系统要求。

## 参考文献


[1] Lorch Markus, Cowles Bob. Conceptual Grid Authorization Framework and Classification[Z]. 2004

[2] Eastlake D, Reagle J, Imamura T, et al. XML Encryption Syntax and Processing[OL]. (2002 - 12 - 10)

[3] Eastlake D, Reagle J, Solo D, et al. XML - Signature Syntax and Processing[EB/OL]. (2002 - 02 - 12)

[4] Ford Warwick, Hallam - Baker Phillip, Fox Barbara, et al. XML Key Management Specification (XKMS)[EB/OL]. (2001 - 03 - 30)[2007 - 8 - 30]. http://www.w3.org/TR/2001/NOTE - xkms - 20010330/

[5] Hallam - Baker Phillip, Mysore Shivaram H. XML Key Management Specification (XKMS 2.0)[EB/OL]. (2005 - 06 - 28)[2007 - 8 - 30]. http://www.w3.org/TR/2005/REC - xkms2 - 20050628

[6] IBM, Microsoft. Security in a Web Services World: A Proposed Architecture and Roadmap[EB/OL]. (2002 - 04 - 07)[2007 - 8 - 23]

[7] Cantor Scott, Kemp John, Philpott Rob, et al. Assertions and Protocols for the OASIS Security Assertion Markup Language[EB/OL]. (2005 - 03 - 15)[2007 - 9 - 3]. http://www.oasis - open.org/committees/security/

[8] Tim Moses. OASIS eXtensible Access Control Markup Language [EB/OL]. (2005 - 02 - 01)[2007 - 8 - 24]. http://www.oasis - open.org/committees/tc_home.php? wg_abbrev = xacml

[9] 傅鹤岗, 李 竞. 基于属性的 Web 服务访问控制模型[J]. 计算机科学. 2007, 34(5): 111 - 114

[10] 杨怀洲, 李增智. 基于 Web Service 的安全业务体系结构的设计[J]. 计算机工程,2005, 31(20): 146 - 148

[11] 韩 涛, 郭荷清. Web 服务安全模型的研究与实现[J]. 计算机工程,2006, 32(10): 130 - 131

[12] 裴艳琴, 杨寿保, 房向明,等. 基于 SAML 的网格策略部署和认证机制[J]. 计算机工程,2007, 33(1): 170 - 172


# Double access control model of web service based on distributed simulation system


Chen Xueqin, Wu Huizhong, Zhu Yaoqin

(*School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China*)



[**Abstract**] As web service is taken up to develop distributed simulation system increasingly, the separation between functions and resources is obvious. This exposes many simulation resources to the wide area network, which are difficult for traditional and single access control models to deal with. A double access control based on attributes is proposed to achieve the access control of application function and simulation resources. The process flow of the access control is described in detail. Also, the model is realized and applied in the prototype of simulation based on acquisition system. The performance of resource access control realized on XACML is analyzed and tested, which proves the feasibility of the model.



[**Key words**] Web service security; double access control; distributed simulation system