Journal of Computer Applications
计算机应用,2016,36(7):1834 - 1840
ISSN 1001-9081
CODEN JYIIDU
2016-07-10
http://www.joca.cn

文章编号:1001-9081(2016)07-1834-07
DOI:10.11772/j.issn.1001-9081.2016.07.1834

# 云系统中多域安全策略规范与验证方法

蔡　婷[1*],蔡　宇[1],欧阳凯[2]

(1.重庆邮电大学移通学院 计算机系,重庆 401520;　2.华中科技大学 计算机学院,武汉 430074)

(* 通信作者电子邮箱 ct_dolphin@163.com)

**摘　要**:为了有效管理云系统间跨域互操作中安全策略的实施,提出一种适用于云计算环境的多域安全策略验证管理技术。首先,研究了安全互操作环境的访问控制规则和安全属性,通过角色层次关系区分域内管理和域间管理,形式化定义了基于多域的角色访问控制(domRBAC)模型和基于计算树逻辑(CTL)的安全属性规范;其次,给出了基于有向图的角色关联映射算法,以实现domRBAC角色层次推理,进而构造出了云安全策略验证算法。性能实验表明,多域互操作系统的属性验证时间开销会随着系统规模的扩大而增加。技术采用多进程并行检测方式可将属性验证时间减少70.1%~88.5%,其模型优化检测模式相比正常模式的时间折线波动更小,且在大规模系统中的时间开销要明显低于正常模式。该技术在规模较大的云系统安全互操作中具有稳定和高效率的属性验证性能。

**关键词**:云系统;多域;访问控制;安全互操作;策略;验证

**中图分类号**:TP309.2　　**文献标志码**:A

# Specification and verification method for security policy in multi-domain cloud systems

CAI Ting[1*], CAI Yu[1], OUYANG Kai[2]

(1. *Department of Computer, College of Mobile Telecommunications, Chongqing University of Posts and Telecommunications, Chongqing* 401520, *China*;
2. *School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan Hubei* 430074, *China*)

**Abstract:** To effectively manage the enforcement of secure policies during the cross-domain interoperation among cloud systems, a management technique applied for the verification of multi-domain cloud policies was proposed. First, both the access control policies and security properties under secure inter-operation environments were studied, the intra-domain administration was distinguished from inter-domain administration according to role hierarchies, and a multi-domain Role Based Access Control (domRBAC) model and specifications for the security properties based on Computation Tree Logic (CTL) were formally defined. Next, a role-to-role mapping algorithm derived from the graph theory was proposed, to depict the reasoning for domRBAC hierarchies, and a verification algorithm of security policies for cloud systems was further constructed. The simulation results show that, the time cost of security policy verification for multi-domains increases with the expansion of the size of the system. Multi-process parallel detection mode can reduce the time of policy verification from 70.1% to 88.5%, and compared to the normal mode, the model optimized detection mode fluctuates smaller in time lines, and the time overhead is significantly lower for large-scale systems. Therefore, the proposed technique has stable performance and high efficiency to be used in secure interoperation of large-scale cloud systems.

**Key words:** cloud system; multi-domain; access control; secure interoperation; policy; verification

## 0　引言

云计算(Cloud Computing)是一种基于因特网的全新商业计算模式,通过广泛的网络带宽接入技术为各类用户提供多租户、可扩展、弹性、按需支付以及可配置的资源,因商业经济效益的驱动得到迅速发展[1]。近年来,各种云计算系统层出不穷:公有云、私有云、混合云、社区云[2]等,然而它们大多是有界的单托管域系统,随着跨域性、动态性资源访问和数据共享需求的扩大,基于多管理域协作的云系统安全策略的实施成为目前工业界和学术界的关注热点[3]。在云计算系统的多域协作研究领域,如何实现访问控制、如何确保跨域互操作的安全性以及如何利用模型检测技术验证安全策略,是3个基本问题。由这3个基本问题延伸出3个重要的研究方向,即访问控制、安全互操作和模型检测。

在访问控制方面,目前的主流方案是基于角色的访问控制(Role-Based Access Control, RBAC)模型。文献[4]指出RBAC支持多种访问控制规则,具有很好的模型抽象和概括能力。文献[5-6]认为RBAC中角色、用户权限的映射关系与实际企业的组织架构层次相对应,适用实际应用环境且易于系统维护和交互管理。文献[7]提出并基于实验证明RBAC较好地解决了云计算应用环境下的企业问题。文献[8]研究了RBAC在云计算系统的现状,认为模型作为云的

收稿日期:2015-12-28;修回日期:2016-03-14。　基金项目:重庆市本科高校"三特行动计划"特色专业建设项目(渝教高(2013)49号);重庆市教委科学技术研究项目(KJ1502002,KJ1502003);重庆市高等教育学会2015—2016年度高等教育科学研究课题(CQGJ15203B);重庆市教育科学"十二五"规划高等教育质量提升专项成果(2015-GX-086)。

作者简介:蔡婷(1984—),女,湖北广水人,讲师,硕士,主要研究方向:互联网计算、网络安全结构与控制;　蔡宇(1979—),男,河南郑州人,讲师,硕士,主要研究方向:云计算、信息安全;　欧阳凯(1978—),男,四川成都人,副教授,博士,主要研究方向:网络安全控制、安全操作系统。

基础设施关键技术适用于多域环境,并且已经成功地应用在医疗、股票和社交网络中。同时,RBAC 模型被大多数主流的云平台所采用,如: OpenStack、Xen、Windows Azure 等。文献[9]提出使用中间件实现多域系统安全策略,并结合医疗案例证实访问控制策略在多域的云计算系统中具有可行性。

在安全互操作方面,大规模分布式环境促使越来越多单托管域的企业联合起来,形成多管理域协作环境成为趋势。文献[10]指出多域安全策略异构问题是各类云系统协作进程中的重要挑战,这个过程要既能有效地支持跨域互操作又能够保证安全。文献[11]通过研究相关文献,提出角色委托机制、映射机制、信任机制和策略合成机制等,是目前学术界涉及的几个研究方向。其中,基于 RBAC 模型的角色映射机制是安全互操作问题的研究热点,例如: 文献[12]为实现分布环境的跨域访问,提出利用一种基于动态角色转换的策略来构建域间访问控制规则和属性约束; 文献[13]在 RBAC 基础上,建立域-域之间的角色映射关系,采用直接转换方式实现域间角色关联以保障单个自治域的安全。

综合上述方案,在 RBAC 策略域的互操作过程中的授权管理与访问控制问题在一定程度上得到了解决。然而,在实施过程中仍然可能出现违反域内安全约束和自治性问题。文献[14]仿真模拟了动态协同环境中安全策略的一致性维护,在定义和维护域间安全访问控制策略方面进行了有益尝试,但并未给出工具检测方法的形式化定义,且缺乏有效的域间访问控制策略的集成方案。

在安全策略验证方面,目前已存在多种模型检测技术[15-17]。这些技术的基本思想是: 用形式化建模语言描述待验证的安全策略系统模型,用时态逻辑公式描述待验证的安全属性,然后将它们输入到检测工具中完成验证。例如,文献[15]提出一种通用访问控制属性验证模型,它能够维护各种静态、动态访问控制的安全约束,并且通过组合验证方式提供测试用例以检测模型和策略规则的一致性; 同时生成基于扩展的访问控制标记语言(eXtensible Access Control Markup Language, XACML)访问控制认证策略,其中 XACML2.0 或 XACML 3.0 已经成为目前协同系统中策略规则的规范化描述语言。文献[16]提出使用黑盒模型检测技术来验证待检测的访问控制属性。文献[17]给出一种访问控制策略检测工具,该工具提供了设置访问控制策略和属性规则的图形化用户界面,可以通过符号模型检测器(Symbolic Model Verifier, SMV)进行访问控制策略的一致性验证。此外,研究还提供了完整的测试工具包以及生成 XACML 语言形式的策略输出。

综上所述,访问控制、安全互操作和模型检测之间是互相制约、相辅相成的,因此,研究一种有效的多域安全策略验证管理技术来实现上述功能具有重要的意义。从公开的国内外文献中还没有发现将上述三者统一起来进行形式化研究并转化应用的成果,类似的研究工作也甚少,且不具有普适性。例如,文献[18]提出一种面向网格系统中分布式访问控制策略的管理方法,研究不同策略行为的表现形式并给出了相应的安全策略验证方案。然而,由于缺乏对于安全互操作问题的关注,其系统模型存在严重的跨域访问安全风险。同时性能评估结果表明,该方案仅适用于小规模的分布式系统、只支持数目相对较小的安全策略的验证。

因此,本文提出了一种适用于云计算系统的多域安全策略验证管理技术,可以在大规模的安全互操作环境中实现形式化定义访问控制规则、规范安全属性和验证安全策略。实现过程表明,该技术通过引入 RBAC 角色层次推理,具有强大的角色关系表达能力,其形式化定义了 RBAC 规则表达式和属性命题,并进一步提出了安全策略验证算法,在大规模安全域模拟实验中显示出更强的通用性和可行性。

## 1　预备知识

简单介绍安全互操作和模型检测的相关理论,RBAC 模型的基础理论请读者自行参阅文献[4],文中不再赘述。

### 1.1　安全互操作

在多域系统中,安全互操作要兼顾自治性和安全性两大原则[3,12]。其中自治性原则是指如果一个访问请求在单个管理域系统中被允许,那么它在安全互操作中也必须被允许; 安全性原则是指如果一个访问请求在单个管理域系统中被禁止,那么它在安全互操作中也须被禁止。在基于 RBAC 模型的安全互操作系统中,域间联合所增加的角色继承关系可能会造成本地安全策略的违反问题,而这种违反约束的行为可以通过相关的策略检测而被预先发现,提前避免安全风险。安全互操作属性有环继承、权限提升、职责分离(Separation of Duty, SoD)原则和自治性等[19]。下面,在给出上述安全属性定义之前,先进行如下约定。

1) $r_1 \rightarrow r_2$,表示角色 $r_1$ 继承角色 $r_2$ 的权限。

2) 如果角色 $r_k$ 属于域 $d_i$ 的角色,则表示为 $d_i r_k$; 同理,$d_i u_t$ 表示域 $d_i$ 的用户 $u_t$,$d_i p_w$ 表示域 $d_i$ 的权限 $p_w$。

定义 1　继承环属性。

在域间互操作过程中,由于新的角色映射关系的引入,角色层次之间形成了环状结构的继承关系,导致下级角色非法拥有了上级角色权限,这种情况称为继承环,记为: $d_i r_j >> d_i r_k$。如图 1(a)所示,在域 $d_i$ 中,用户 $d_i u_t$ 被指派给角色 $d_i r_k$,则用户 $d_i u_t$ 同时获得到了它的上级角色 $d_i r_j$ 的权限。

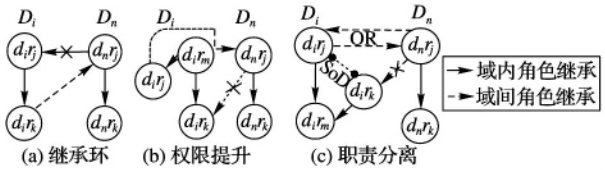


图 1　3 种安全属性

定义 2　权限提升属性。

在域间互操作过程中,新的角色关联关系导致以前没有关联的角色之间形成某种继承关系,使得角色获取到更大的权限,这种情况称为权限提升,记为: $d_i r_j \geqslant d_i r_k$。如图 1(b)所示,域 $d_i$ 用户 $d_i u_t$ 被指派给角色 $d_i r_j$,用户 $d_i u_t$ 在获得角色 $d_i r_j$ 权限的同时还获得了角色 $d_i r_k$ 的权限,即使用户 $d_i u_t$ 与角色 $d_i r_k$ 之间并不存在直接指派关系。

定义 3　职责分离属性。

如果域 $d_i$ 用户 $d_i u_t$(或者域 $d_n$ 用户 $d_n u_t$)由于域间角色映射关系,使得它可以获取或在会话中激活存在 SoD 约束的两互斥角色 $d_i r_j$ 和 $d_i r_k$,那么就违反了 SoD 约束,如图 1(c)所示。本文验证职责分离属性是基于下面两个性质[4]:

1) 如果角色 $r_k$ 和 $r_m$ 之间不存在直接或间接的继承关系,那么 $r_k$ 和 $r_m$ 完全互斥;

2) 如果角色 $r_k$ 和 $r_m$ 完全互斥,那么不存在有任何角色可以同时继承 $r_k$ 和 $r_m$。

**定义4** 自治性属性。

自治性属性要求在域间互操作环境中的访问控制权限不能违反自治管理域的本地操作权限。安全互操作要求平衡自治性和交互性,违反任何单个域的安全策略都是不允许的。

### 1.2 模型检测

模型检测技术是验证安全互操作属性的重要手段,它能够解决访问控制模型的通用属性验证问题。早在20世纪80年代,基于时序逻辑的模型检测技术[20]就被广泛关注,其原理如图2描述:假设,$M$ 表示状态迁移系统,$F$ 表示模态时序逻辑公式,将"系统是否具有所期望的性质"转化成数学问题来描述,即"$M$ 是否是公式 $F$ 的一个模型",记为 $M \models F$?。

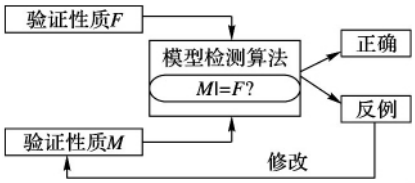


图2 模型检测的基本流程

模型检测过程主要包括系统建模、建立系统性质规范和执行验证3个过程。其中,系统建模主要是建立与系统相对应的迁移系统或Kripke[16]结构,用来描述系统方案的动态行为;系统性质规范的建立要求统一系统性质的表达形式,多数会使用计算树逻辑(Computation Tree Logic,CTL)、线性时态逻辑(Linear Temporal Logic,LTL)等属性描述语言来规范表达;执行验证环节可以采用方便的自动验证模式,由模型检测器完成。

目前,存在大量支持模型检测技术应用的模型检测工具,如:SMV、简单进程元语言解释器(Simple Promela Interpreter,SPIN)[21]、改进符号模型检测器(New Symbolic Model Verifier,NuSMV)[22]和Uppaal[23]等。本文后续的研究工作选用NuSMV这款开放架构的模型检测器。

## 2 云安全策略模型

在本章中,主要完成两方面的工作:

1) 针对云系统中RBAC方案不能有效地解决不同云托管域的策略集成问题,引入域内管理和域间管理两类角色层次关系,对传统的适用于单域的RBAC模型进行重定义,从而建立一种基于多域的角色访问控制(multi-domain Role Based Access Control,domRBAC)模型;

2) 给出基于CTL语言的通用访问控制模型转换方法[17],并对访问控制规则、安全属性和迁移系统的表达进行了规范。

### 2.1 domRBAC模型

本文在ANSI INCITS 359-2004 RBAC[4]的基础上,综合考虑了系统功能和审查功能,给出如下形式化定义。

#### 2.1.1 基本元素

1) $USERS$、$ROLES$、$OPS$、$OBS$ 分别表示用户、角色、操作、对象的集合。

2) $UA \subseteq USERS \times ROLES$,表示用户与角色之间多对多的分配关系。

3) $PRMS = 2^{(OPS \times OBS)}$,表示权限的集合。

4) $PA \subseteq PRMS \times ROLES$,表示权限与用户之间多对多的分配关系。

5) $Op(p: PRMS) \rightarrow \{op \subseteq OPS\}$,表示权限与操作之间的对应关系,指明为操作集分配的权限集 $p$。

6) $Ob(p: PRMS) \rightarrow \{ob \subseteq OBS\}$,表示权限与对象之间的对应关系,指明为对象集分配的权限集 $p$。

#### 2.1.2 域内角色层次

1) $assigned\_users: SU_{d_i}(d_i r_k: ROLES) \rightarrow 2^{USERS}$,表示域 $d_i$ 中角色 $d_i r_k$ 与用户集 $USERS$ 之间的映射关系,即:$SU_{d_i}(d_i r_k) = \{d_i u_t \in USERS \mid (d_i u_t, d_i r_k) \in UA\}$。

2) $assigned\_permissions: SP_{d_i}(d_i r_k: ROLES) \rightarrow 2^{PRMS}$,表示域 $d_i$ 中角色 $d_i r_k$ 与权限集 $PRMS$ 之间的映射关系,即:$SP_{d_i}(d_i r_k) = \{d_i p_w \in PRMS \mid (d_i p_w, d_i r_k) \in PA\}$。

3) $RH_{d_i} \subseteq ROLES \times ROLES$,表示域 $d_i$ 中角色之间继承关系的偏序集合,记为 $\geq$。若 $d_i r_k \geq d_i r_m$,那么 $d_i r_m$ 的权限集都是 $d_i r_k$ 的权限集,且 $d_i r_k$ 的用户集则都是 $d_i r_m$ 的用户集,即:$d_i r_k \geq d_i r_m \Rightarrow UP_{d_i}(d_i r_m) \subseteq UP_{d_i}(d_i r_k) \wedge UU_{d_i}(d_i r_k) \subseteq UU_{d_i}(d_i r_m)$。

4) $authorized\_users: UU_{d_i}(d_i r_k: ROLES) \rightarrow 2^{USERS}$,表示域 $d_i$ 中角色 $d_i r_k$ 与域内角色层次用户集 $USERS$ 之间的映射关系,这种映射只考虑角色 $d_i r_k$ 与域内的其他角色之间的继承关系,即:$UU_{d_i}(d_i r_k) = \{d_i u_t \in USERS \mid d_i r_m \geq d_i r_k, (d_i u_t, d_i r_m) \in UA\}$。

5) $authorized\_permissions: UP_{d_i}(d_i r_k: ROLES) \rightarrow 2^{PRMS}$,表示域 $d_i$ 中角色 $d_i r_k$ 与域内角色层次权限集 $PRMS$ 之间的映射关系,这种映射只考虑角色 $d_i r_k$ 与域内的其他角色之间的继承关系,即:$UP_{d_i}(d_i r_k) = \{d_i p_w \in PRMS \mid d_i r_k \geq d_i r_m, (d_i p_w, d_i r_m) \in PA\}$。

#### 2.1.3 域间角色层次

1) $RH \subseteq ROLES \times ROLES$,表示域间角色之间继承关系的偏序集合,记为 $\geq$。若 $d_i r_k \geq d_j r_m$,那么 $d_j r_m$ 的权限集都是 $d_i r_k$ 的权限集,且 $d_i r_k$ 的用户集都是 $d_j r_m$ 的用户集。即:$d_i r_k \geq d_j r_m \Rightarrow UP(d_j r_m) \subseteq UP(d_i r_k) \wedge UU(d_i r_k) \subseteq UU(d_j r_m)$。

2) $authorized\_users: UU(d_i r_k: ROLES) \rightarrow 2^{USERS}$,表示角色 $d_i r_k$ 与域间角色层次用户集 $USERS$ 之间的映射关系,这种映射的集合既包括 $d_i r_k$ 与域内角色之间的继承关系,又包括 $d_i r_k$ 与外域角色之间的继承关系,即:$UU(d_i r_k) = UU_{d_i}(d_i r_k) \cup \{d_j u_t \in USERS \mid d_j r_m \geq d_i r_k, (d_j u_t, d_j r_m) \in UA\}$。

3) $authorized\_permissions: UP(d_i r_k: ROLES) \rightarrow 2^{PRMS}$,表示角色 $d_i r_k$ 与域间角色层次权限集 $PRMS$ 之间的映射关系,这种映射关系既包括 $d_i r_k$ 与域内角色之间的继承,又包括 $d_i r_k$ 与外域角色之间的继承,即:$UP(d_i r_k) = UP_{d_i}(d_i r_k) \cup \{d_j p_w \in PRMS \mid d_i r_k \geq d_j r_m, (d_j p_w, d_j r_m) \in PA\}$。

#### 2.1.4 谓词

考虑到时序逻辑语言中缺乏关系算子,如:□ 和 ≥。下面,补充一些对应谓词的定义。

1) $IR(r_k, r_m)$ 表示两角色间存在(域间或域内)直接继承关系,即:$IR(r_k, r_m) = \text{true} \Leftrightarrow r_k \square r_m$。其中,符号 □ 表示直接继承关系。

2) $MR_{d_i}(d_i r_k, d_i r_m)$ 表示域 $d_i$ 角色层次中的两角色间存

在一种(域间或域内)直接的或者间接的继承关系,即: $MR_{d_i}(d_ir_k,d_ir_m)=\text{true}\Leftrightarrow d_ir_k\geq d_ir_m$。

3) $RP(r_k,r_m)$ 表示对于存在直接继承关系的两角 $r_k$, $r_m(r_k\square r_m)$,角色 $r_k$ 的分配权限集是角色 $r_m$ 权限集的子集,即: $RP(r_k,r_m)=\text{true}\Leftrightarrow IR(r_k,r_m)\wedge SP_{d_i}(r_k)\subseteq UP(r_m)$。

4) $IB_{d_i}(d_ir_k,d_ir_m,r_n)$ 表示如果任意域角色 $r_n$ 是所在域 $d_i$ 中角色 $d_ir_k$ 和角色 $d_ir_m$ 的上级角色,那么, $r_n$ 的权限集则同时包括了 $d_ir_k$ 的权限集和 $d_ir_m$ 的权限集,即: $IB_{d_i}(d_ir_k,d_ir_m,r_n)=\text{true}\Leftrightarrow SP_{d_i}(d_ir_k)\cup SP_{d_i}(d_ir_m)\subseteq UP(r_n)\wedge r_n\geq d_ir_k\wedge r_n\geq d_ir_m$。

5) $BA(d_ir_k)$ 表示角色 $d_ir_k$ 与域内角色层次中权限集的映射关系,是 $d_ir_k$ 与域间角色层次中权限集的映射关系的子集,即: $BA(d_ir_k)=\text{true}\Leftrightarrow UP_{d_i}(d_ir_k)\subseteq UP(d_ir_k)$。

### 2.2　转换系统

本文采用 CTL 时序逻辑来对有关的安全策略进行规范,如: 访问控制规则、安全属性和变迁系统。

在 CTL 语言中,前缀路径量词可以断言关于线性时序算子的任意组合。据此,本文规定使用通用路径量词 $\forall$ 表示"对所有路径",使用线性时序算子 $\square$ 表示"现在和以后所有状态",使用线性时序算子 $\diamond$ 表示"现在或以后某一状态"。另外,规定时序模式 $\forall\square\Phi$ 表示不变的 $\Phi$,时序模式 $\forall\diamond\Phi$ 表示可变的 $\Phi$,其中, $\Phi$ 是一个状态公式。

**定义 5**　一条 domRBAC 规则是形如"if $c$ then $d$"的命题,其中,约束 $c$ 是一个关于决策许可 $d$ 的谓词表达式($r$, $UP(r)$),因此,由一系列规则组成的 domRBAC 策略,可以表示成形如 $c(r,UP(r))$ 的这种逻辑表达式形式。

**定义 6**　一个 domRBAC 访问控制属性 $p$ 是形如"$b\rightarrow d$"的公式,其中,访问权限许可 $d$ 的结果取决于量化谓词 $b$ 与($r$, $UP(r)$)之间的映射关系,其归约关系 $\rightarrow$ 描述了系统内部的推理方式。

**定义 7**　迁移系统 $TS$ 是一个四元组 $(S,Act,\delta,i_0)$,其中:

1) $S$ 是有限状态的集合, $S=\{Permit,Deny\}$;

2) $Act$ 是活动的集合, $Act=\{(r_1,UP(r_1)),(r_2,UP(r_2)),\cdots,(r_n,UP(r_n))\}$;

3) $\delta$ 是状态转移关系,且 $\delta: S\times Act\rightarrow S$;

4) $i_0\in S$ 是初始状态。

根据定义 6,访问控制属性 $p$ 可以被表示成迁移系统 $TS$ 的命题,如 $p: S\times Act^2\rightarrow S$,因此,domRBAC 策略可以对应地转换成逻辑公式: $p=(S_i*(r_1,UP(r_1))*(r_2,UP(r_2))*\cdots*(r_n,UP(r_n)))\rightarrow d$,其中, $p\in P$, $P$ 代表属性集合,并且 $*$ 是 CTL 中的布尔算子。此外,domRBAC 模型的功能规则对应于转换系统 $TS$ 的转换关系 $\delta$,因此,将 domRBAC 访问控制属性表示为时态逻辑表达式(即时态规范),就可以断言属性 $p$ 在 $TS$ 下是否可满足,即验证 $TS\models\forall\square(b\rightarrow\forall\diamond d)$ 是否为真。

### 2.3　属性规范

结合前面 2.1 节内容,下面给出继承环属性、权限提升属性、职责分离属性以及自治性安全属性的时态逻辑定义。

**定义 8**　继承环属性为:

$$TS_{\text{domRBAC}}\models\forall\square(RP(d_ir_j,d_ir_k)\rightarrow\forall\diamond Deny)\tag{1}$$

其中 $d_ir_j$, $d_ir_k$ 表示域 $d_i$ 中的两个角色。通过验证命题 $RP(d_ir_j,d_ir_k)\rightarrow\forall\diamond Deny$ 是否满足 $TS_{\text{domRBAC}}$ 中的不变式,来检测角色 $d_ir_j$ 是否存在环状继承。

**定义 9**　权限提升属性为:

$$TS_{\text{domRBAC}}\models\forall\square((\neg MR_{d_i}(d_ir_j,d_ir_k)\wedge RP(d_ir_j,d_ir_k))\rightarrow\forall\diamond Deny)\tag{2}$$

其中 $d_ir_j$ 是用户 $d_iu_t$ 对应的指派角色。通过验证命题 $(\neg MR_{d_i}(d_ir_j,d_ir_k)\wedge RP(d_ir_j,d_ir_k))\rightarrow\forall\diamond Deny$ 是否满足不变式 $TS_{\text{domRBAC}}$,来检测角色 $d_ir_j$, $d_ir_k$ 之间是否因为域间映射关联导致用户 $d_iu_t$ 的权限提升。

**定义 10**　职责分离属性为:

$$TS_{\text{domRBAC}}\models\forall\square((d_ir_j\in d_irs_w\wedge d_ir_k\in d_irs_w\wedge(RP(d_ir_j,d_ir_k)\vee RP(d_ir_k,d_ir_j)\vee IB_{d_i}(d_ir_j,d_ir_k,r_m)))\rightarrow\forall\diamond Deny)\tag{3}$$

鉴于 SoD 属性是基于角色对实现的,这就需要检测互斥角色对的最小数量的约束关系: $(d_irs,n)\in SSD$,其中, $n\geqslant 2$ 且 $d_irs$ 代表一个角色集。同样地,可以等价地表示成二项系数 $_{|d_irs|}C_2\equiv\begin{pmatrix}|d_irs|\\2\end{pmatrix}\equiv\frac{|d_irs|!}{2!(|d_irs|-2)!}$。

**定义 11**　自治性属性为:

$$TS_{\text{domRBAC}}\models\forall\square(BA(d_ir_k)\rightarrow\forall\diamond Permit)\tag{4}$$

在互操作中,通过检测域 $d_i$ 中角色 $d_ir_k$ 的所有指派权限和角色层次映射生成权限是否被保护,来验证自治性属性。

## 3　技术实现

本章讨论云系统多域安全策略验证技术实现问题。首先,提出一种基于图论的角色关联(角色-角色)映射算法,该算法通过引入 RBAC 角色层次推理来实现对系统模型中角色层次关系的准确模拟。该算法的核心思想是,用稀疏图数据结构表示角色层次关系,用链表替代传统矩阵模拟角色层次,以获取更高的属性验证性能。其次,给出了基于多域的云安全策略验证算法。下面,先给出实现部分的相关定义。

**定义 12**　$G=(V,E)$ 是一个表达域间角色层次的有向图,其中 $V(V\subseteq ROLES)$ 代表一组有限、非空的角色顶点集合, $E$ 代表图中有向边的集合,并且,每条有向边都是相关两角色顶点的一对序偶 $(d_ir_m,d_jr_n)$,其中两角色顶点的关系为 $d_ir_m\geq d_jr_n$。

**定义 13**　图 $G$ 中的一条路径是指由 $n-1$ 条有向边所构成的序列集合 $\{(d_ir_1,d_ir_2),(d_ir_2,d_ir_3),\cdots,(d_ir_{n-1},d_ir_n)\}$,连接从角色顶点 $d_ir_1$ 到角色顶点 $d_ir_n$,一条路径代表了两角色顶点 $d_ir_1$ 和 $d_ir_n$ 之间的间接继承关系。

**定义 14**　图 $G$ 的邻接表是列表 $|V|$ 的一个数组 $L$,图 $G$ 中的每个角色顶点都被包含在 $V$ 集合里面。对于每个角色顶点 $d_ir_m$ 来说,都存在一个指针 $L_{d_ir_m}$ 指向一个涵盖与 $d_ir_m$ 相邻接的所有角色顶点的链接表。本文用 $A_G$ 表示图 $G$ 的邻接表,用 nil 指针表示一个链表的终止。

**定义 15**　$G^*=(V,E^*)$ 是图 $G=(V,E)$ 的传递闭包,其中,当且仅当图 $G$ 中存在一条从顶点 $u$ 到顶点 $v$ 的路径时, $E^*$ 集合中包含有一条边 $edge(u,v)$。本文用 $T_G$ 表示一个基于邻接表存储的有向图 $G=(V,E)$ 的传递闭包列表。

基于图论的角色关联映射算法的结构如算法 1 所示。这里, $T_G$ 为算法返回的生成结果,期间采用的改进 Warshall 算法的相关信息可参考文献[24]。在角色关联映射算法中,第 1

步，根据 domRBAC 规则生成有向图 $G=(V,E)$ 的邻接表 $A_G$，这个过程可以利用如文献[22]中提到的解析器(Simple API for XML，SAX)进行自动生成；第 2 步，根据 $A_G$ 计算图 $G$ 的传递闭包列表 $T_G$，本文采用一种时间复杂度为 $O(|V||E|)$ 的改进传递闭包算法[24]。

算法 1　基于图论的角色关联映射算法(如图 3)。

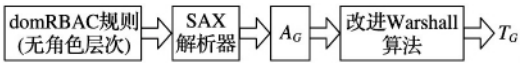


图 3　基于图论的角色关联映射算法

示例 1　如图 4 所示是一个定义了两个域 $d_1$，$d_2$ 之间互操作的域间访问控制策略应用场景。其中，在管理域 $d_1$ 中有角色 $d_1r_a$，$d_1r_b$，$d_1r_c$，$d_1r_d$，$d_1r_e$，角色 $d_1r_a$ 继承 $d_1r_b$ 的所有权限并间接继承 $d_1r_e$ 的权限，角色 $d_1r_c$ 继承 $d_1r_d$ 的所有权限并间接继承 $d_1r_e$ 的权限，并且角色 $d_1r_b$ 和角色 $d_1r_c$ 之间还存在 SSD 约束；在管理域 $d_2$ 中有两个角色 $d_2r_f$，$d_2r_g$，角色 $d_2r_f$ 继承 $d_2r_g$ 的全部权限。此外，域 $d_1$ 和 $d_2$ 之间的域间继承关系定义如下：

1) 角色 $d_1r_b$ 继承角色 $d_2r_g$；

2) 角色 $d_2r_g$ 继承角色 $d_1r_c$。

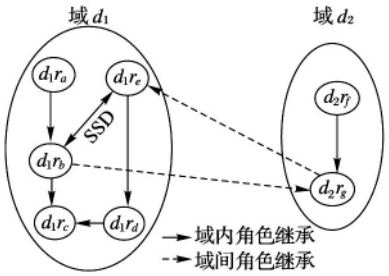


图 4　多域访问控制策略

如图 5 所示是 $A_G$ 和 $T_G$ 的生成结果，具体的计算过程如下所示。

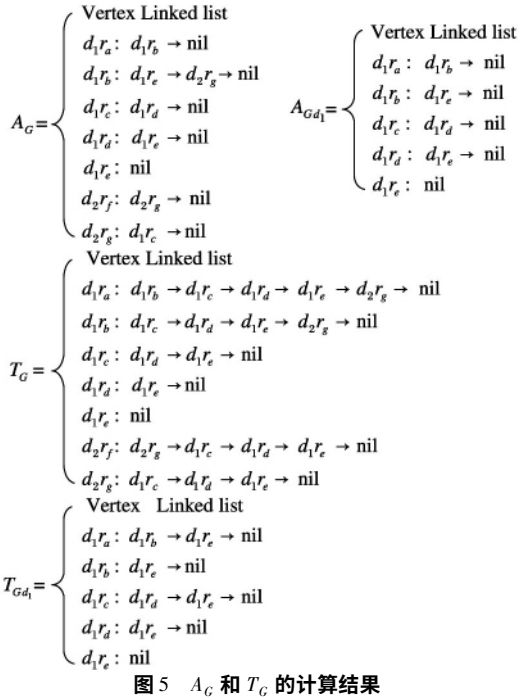


图 5　$A_G$ 和 $T_G$ 的计算结果

首先，利用 BOOST C++ 程序库[25]中的 Boost Graph 中 adjacency_list 类，生成一个通用的以邻接表 $A_G$ 结构存储的有向图 $G$；其次，利用 Boost Graph 库中的 transitive_closure() 函数，将图 $G$ 输入并转换生成传递闭包结构列表 $T_G$。

图 5 中的 $AG_{d1}$ 和 $TG_{d1}$ 分别表示管理域 $d_1$ 的邻接表和传递闭包列表。根据定义 11，$TG_{d1}$ 可以作为一种待检测的安全属性，用于验证原始单个管理域在与多域的互操作过程中是否违反了自治性原则。

基于多域的云安全策略验证算法的结构如算法 2 和图 6 所示。在该算法中，首先利用算法 2，将根据云用户访问需求生成的 domRBAC 规则 XML 文件，经过解析器 SAX 生成记录域间角色关联关系的 $A_G$ 和 $T_G$；然后，利用迭代器(Iterator)，根据算法 3，迭代生成新的 XML 文件，具体包括原有的访问控制规则、新增角色关系描述规则以及待验证的安全属性。由于这些 XML 文件是经过规范的逻辑程序，因而可以装载进入检测系统直接计算；最后，NuSMV 系统可高效地计算它并返回查询结果 $R$：

1) 如果 $R$ = true，即 $TS_{\text{domRBAC}} \models p$ 为真。说明属性 $p$ 在 $TS$ 下是可满足。

2) 如果 $R$ = FALSE，即 $TS_{\text{domRBAC}} \models p$ 为假。说明属性 $p$ 在 $TS$ 下是不可满足。

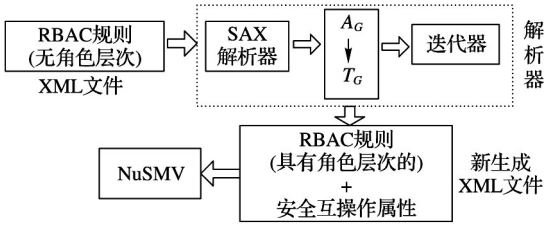


图 6　云安全策略验证算法

算法 2　云安全策略验证算法。

算法 3　规则和属性生成算法。

```
procedure ITERATOR_SKELETON(T_G)
  for all vertex dr_i ∈ T_G
    for all adjacent vertex dr_j
      生成时态规范的规则和属性
end procedure
```

## 4　性能评测

下面对本文提出的技术进行实验性能评估。实验说明如下：首先，利用 NetworkX 工具生成云端用户的访问控制请求，作为解析器的输入数据。其中，NetworkX 是一款用 Python 语言开发的图论和复杂网建模工具，能够提供 gnc_graph() 函数动态生成用户请求和对应权限。其次，使用 domRBAC 模拟器模拟云托管域的角色指派。

假设分别有 5，10，15，20 个云托管域，其中每个托管域中含有 50 个角色。在此基础上，实验设计模拟 4 个不同规模大小的域间互操作环境，分别是具有 250，500，750，1 000 个角色的云协同计算环境。系统的运行环境为 Windows Server 2003 R2 操作系统，CPU 版本为 Intel Core2 3.0 GHz，内存为 4 GB DDR2，开发语言为 C++。

实验提供了一系列有关安全属性验证时间的定量结果。如表 1 所示，给出了解析器和 NuSMV 验证器的实验数据，其中，$T_G$ 的数量对应访问控制规则的数目。从表 1 看出，执行时间：1# < 2# < 3# ≪ 4#，说明规模越大的系统，其属性验证的时间开销也越大。通过分析可知，系统中安全属性的检测耗时会随着安全属性数目以及 domRBAC 规则数量的增加而增

大,这是因为属性数和规则数决定了检测器中有序二叉决策图( Binary Decision Diagram, BDD) 的可到达状态数,直接影响了状态判断次数,那么,如何降低系统规模对于属性验证时间的影响? 可以考虑采用并行检测的方式进行属性验证,并且,表 1 的检测数据是在 NuSMV 模型验证器的正常模式下实验采集的,如果采用优化模式,性能会因增加 3 个参数设置而相对提高,因为正常检测模式是不包含任何额外的命令行参数的。

表 1　评估数据汇总

| 系统序号 | 角色数 | $T_C$数量 | 安全属性数 | 可到达状态数 | 执行时间/min |
|---|---|---|---|---|---|
| 1# | 5×50 | 1 049 | 12 419 | $2^{29.2379}$ | <1 |
| 2# | 10×50 | 1 821 | 27 276 | $2^{32.2296}$ | 182 |
| 3# | 15×50 | 3 182 | 52 168 | $2^{33.9811}$ | 1 490 |
| 4# | 20×50 | 3 467 | 76 977 | $2^{35.2249}$ | 5 847 |

基于上述分析,后续实验将围绕两个方面展开设计: 一方面,引入并行检测和优化检测两种实验模式,分模式测试不同规模系统的属性验证时间; 另一方面,分规模测试不同并行进程数的属性验证时间。

如表 2 所示,在 8 个进程并行检测模式下,分别测量了正常模式( N) 和优化模式( O) 的属性验证时间,N 和 O 是指 NuSMV 模型检测器实验时的两种参数设置状态。其中,1#( 5×50) 规模的系统时间太小( <1 min) ,可忽略不计( 表 2 中表示为 "—") 。表 2 中的时间减少率( *Reduction_time*) ,定量描述了多进程并行检测模式对比单进程串行检测模式的执行效率,它的具体计算方法如下:

$$Reduction\_time = (1 - maxT/Single\_process\_time) \times 100\% \quad (5)$$

其中: $maxT$ 表示 $(t_{P_i})_{i=1}^{N}$ 的最大值,并且 $N$ 表示并行执行的进程数; $t_{P_i}$ 表示进程 $P_i(1 \leqslant i \leqslant N)$ 的执行时间; $\sum_{i=1}^{N} t_{P_i}$ 则表示多个进程顺序执行的时耗总和。

表 2　两种模式的性能测试

| 规模模式 | 单进程时间/min | $\sum_{i=1}^{N} t_{P_i}$ /min | $maxT$/min | $minT$/min | 时间减少比例/% |
|---|---|---|---|---|---|
| 1N | <1 | — | — | — | — |
| 1O | <1 | — | — | — | — |
| 2N | 182 | 145 | 21 | 15 | 88.5 |
| 2O | 135 | 181 | 23 | 20 | 82.9 |
| 3N | 1 490 | 1 333 | 210 | 128 | 85.9 |
| 3O | 749 | 1 434 | 224 | 165 | 70.1 |
| 4N | 5 847 | 5 340 | 908 | 360 | 84.5 |
| 4O | 1 508 | 1 970 | 285 | 200 | 81.1 |

如图 7 ~ 9 所示,分别显示了 2#( 10×50) 、3#( 15×50) 、4#( 20×50) 三个规模的系统在正常模式和优化模式下的时间性能。

上述实验结果表明:

1) 并行检测显著地提高了系统的属性验证性能。例如,表 2 中的数据显示,当并行检测进程数为 8 时,在三种规模的系统中,正常模式下 *Reduction_time* 分别为 88.5%、85.9%、84.5%; 优化模式下 *Reduction_time* 分别为 82.9%、70.1%、81.1%。相比单个进程检测模式具有很好的时间性能。

2) 时间随机波动,优化模式比正常模式具有更稳定的属性验证性能。从图 7 ~ 9 中看出,模型验证器 NuSMV 的执行时间在正常模式相比优化模式下显示出了更大的波动性和不可预测性。随着并行进程数的不断增加,系统安全属性的验证时间必然会因角色数量的增加而受到影响。随机波动则是由安全属性的数量、BDD 可到达状态的数量等多因素共同影响所致。

3) 在大规模系统中,优化模式下的属性验证时间开销要明显低于正常模式; 然而,在中小规模系统中,反而是正常模式下的属性验证时间开销明显低于优化模式。例如,图 7 ~ 9,在 2#( 10×50) 和 3#( 15×50) 系统中,存在 $maxT_{\text{normal}} < maxT_{\text{optimized}}$ 和 $minT_{\text{normal}} < minT_{\text{optimized}}$。在 4#( 20×50) 系统中,是 $maxT_{\text{normal}} \gg maxT_{\text{optimized}}$ 和 $minT_{\text{normal}} \gg minT_{\text{optimized}}$,因此,建议在中小规模系统中选择模型正常检测模式,在规模较大的系统中则应该选择模型优化检测模式。

4) $\sum_{i=1}^{N} t_{P_i} \neq Single\_process\_time$,说明在执行相同数目的属性验证时,多个进程顺序执行的时耗总和与单进程执行验证的总时间并不相等。

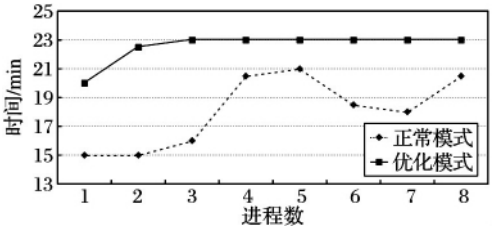

图 7　2#系统的属性并行验证测试

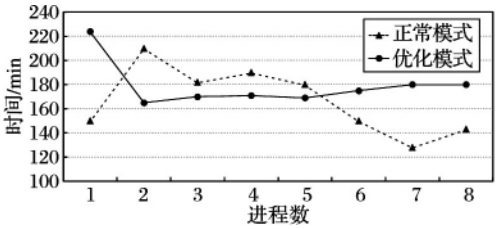

图 8　3#系统的属性并行验证测试

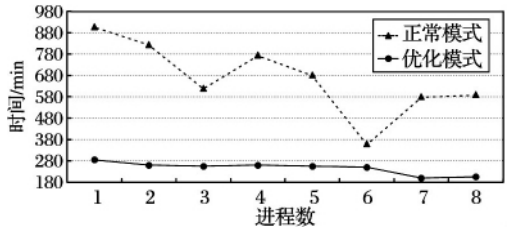

图 9　4#系统的属性并行验证测试

## 5　结语

本文研究了基于访问控制规则和安全互操作属性的策略规范和验证问题,提出了一种适用于云计算系统的多域安全策略验证管理技术。文中的主要工作如下: 1) 提出一种基于多域环境的角色访问控制( domRBAC) 模型; 2) 研究了安全互操作理论,建立了基于 CTL 时序逻辑的转换规范,并给出环继承属性、权限提升属性、职责分离属性以及自治性属性的时态逻辑表达形式; 3) 给出了技术的详细实现,为基于多域的安全策略验证管理提供了一整条工具链。实验结果表明,该技术方案能够较好地实现域间互操作中的安全策略表达、规

范和安全策略验证，在较大规模的云系统中具有稳定性、高效性和可行性。下一步将完善跨域资源使用约束、模型检测算法和访问安全威胁消解等方面的研究，进一步提高云系统中多域安全策略的管理效果。

**参考文献：**


[1] 杨健，汪海航，王剑，等. 云计算安全问题研究综述[J]. 小型微型计算机系统，2012，33(3)：472－479.（YANG J，WANG H H，WANG J，et al. Survey on some security issues of cloud computing [J]. Journal of Chinese Computer Systems，2012，33(3)：472－479.）

[2] ARMBRUST M，FOX A，GRIFFITH R，et al. A view of cloud computing [J]. Communications of the ACM，2010，53(5)：50－58.

[3] 陈华山，皮兰，刘峰，等. 网络空间安全科学基础的研究前沿及发展趋势[J]. 信息网络安全，2015(3)：1－5.（CHEN H S，PI L，LIU F，et al. Research on frontier and trends of science of cybersecurity [J]. Netinfo Security，2015(3)：1－5.）

[4] American National Standard Institute. ANSI INCITS 359-2004，American national standard for information technology-role based access control [S]. New York：American National Standards Institute，2004.

[5] SCHEFER-WENZL S，STREMBECK M. Modeling context-aware RBAC models for business processes in ubiquitous computing environments [C]// Proceedings of the 2012 Third FTRA International Conference on Mobile，Ubiquitous，and Intelligent Computing. Piscataway，NJ：IEEE，2012：126－131.

[6] YAN D，TIAN Y，HUANG J，et al. Privacy-aware RBAC model for Web services composition [J]. The Journal of China Universities of Posts and Telecommunications，2013，20(S1)：30－34.

[7] LI W，WAN H，REN X，et al. A refined RBAC model for cloud computing [C]// Proceedings of the 2012 IEEE/ACIS 11th International Conference on Computer and Information Science. Piscataway，NJ：IEEE，2012：43－48.

[8] 叶春晓，郭东恒. 多域环境下安全互操作研究[J]. 计算机应用，2012，32(12)：3422－3425.（YE C X，GUO D H. Research on secure interoperation in multi-domain environment [J]. Journal of Computer Applications，2012，32(12)：3422－3425.）

[9] MIGLIAVACCA M，PAPAGIANNIS I，EYERS D M，et al. Distributed middleware enforcement of event flow security policy [C]// Proceedings of the 2010 ACM/IFIP/USENIX 11th International Conference on Middleware. Berlin：Springer，2010：334－354.

[10] TAKABI H，JOSHI J B D，AHN G J. Security and privacy challenges in cloud computing environments [J]. IEEE Security & Privacy，2010，8(6)：24－31.

[11] 邹德清，邹永强，羌卫中，等. 网格安全互操作及其应用研究[J]. 计算机学报，2010，33(3)：514－525.（ZOU D Q，ZOU Y Q，QIANG W Z，et al. Grid security interoperation and its application [J]. Chinese Journal of Computers，2010，33(3)：514－525.）

[12] KAPADIA A，AL-MUHTADI J，ROY C，et al. IRBAC 2000：secure interoperability using dynamic role translation [R]. Champaign，IL：University of Illinois at Urbana-Champaign，1999：1－7.

[13] AL-MUHTADI J，KAPADIA A，CAMPBELL R，et al. The A-IRBAC 2000 model：administrative interoperable role-based access control [R]. Champaign，IL：University of Illinois at Urbana-Champaign，2000：1－9.

[14] HU V C，KUHN D R，XIE T. Property verification for generic access control models [C]// EUC'08：Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing. Piscataway，NJ：IEEE，2008，2：243－250.

[15] BRYANS J W，FITZGERALD J S. Formal Engineering of XACML Access Control Policies in VDM++ [M]. Berlin：Springer，2007：1－23.

[16] HU V C，KUHN D R，XIE T，et al. Model checking for verification of mandatory access control models and properties [J]. International Journal of Software Engineering and Knowledge Engineering，2011，21(1)：103－127.

[17] HWANG J H，XIE T，HU V，et al. ACPT：a tool for modeling and verifying access control policies [C]// Proceedings of the 2010 IEEE International Symposium on Policies for Distributed Systems and Networks. Piscataway，NJ：IEEE，2010：40－43.

[18] HWANG J H，ALTUNAY M，XIE T，et al. Model checking grid policies [EB/OL]. [2015-11-28]. http://hwang250.google-code.com/.

[19] SHAFIQ B，JOSHI J B D，BERTINO E，et al. Secure interoperation in a multidomain environment employing RBAC policies [J]. IEEE Transactions on Knowledge and Data Engineering，2005，17(11)：1557－1577.

[20] CLARKE E M，EMERSON E A. Design and synthesis of synchronization skeletons using branching time temporal logic [M]// 25 Years of Model Checking. Berlin：Springer，2008：196－215.

[21] HOLZMANN G J. The model checker SPIN [J]. IEEE Transactions on Software Engineering，1997，23(5)：279－295.

[22] CIMATTI A，CLARKE E，GIUNCHIGLIA F，et al. NuSMV：a new symbolic model checker [J]. International Journal on Software Tools for Technology Transfer，2000，2(4)：410－425.

[23] BERHMANN G，DAVID A，LARSEN K G. A tutorial on UPPAAL [M]// Formal Methods for the Design of Real-Time Systems. Berlin：Springer，2004：200－236.

[24] PAPADIMITRIOU C，SIDERI M. On the Floyd-Warshall algorithm for logic programs [J]. Journal of Logic Programming，1999，41(1)：129－137.

[25] HERB S，ANDREI A. Boost C++ libraries 1.58.0 [EB/OL]. [2015-04-17]. http://www.boost.org/.



**Background**

This work is partially supported by the "Three Special Action Plan" Specialty Construction Project of Universities in Chongqing (Yu Teach High (2013) No. 49)，the Science and Technology Research Project of Education Committee of Chongqing (KJ1502002，KJ1502003)，the Scientific Research Programs in Higher Education of Chongqing Institute of Higher Education (CQGJ15203B)，the Quality Improvement Projects in Higher Education of Chongqing Education Science "Twelfth Five Year Plan" (2015-GX-086).



**CAI Ting**，born in 1984，M. S.，lecturer. Her research interests include Internet computing，network security structure and control.

**CAI Yu**，born in 1979，M. S.，lecturer. His research interests include cloud computing，information security.

**OUYANG kai**，born in 1978，Ph. D.，associate professor. His research interests include network security control，secure operating system.