软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(6):1538−1548 [doi: 10.13328/j.cnki.jos.004997]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种透明的可信云租户隔离机制研究*

石 勇[1], 郭 煜[1], 刘吉强[1], 韩 臻[1], 马 威[1], 常 亮[2]

[1](北京交通大学 计算机与信息技术学院,北京 100044)

[2](广西可信软件重点实验室(桂林电子科技大学),广西 桂林 541004)

通讯作者: 石勇, E-mail: stonefly128@126.com

摘 要: 租户隔离是云计算能被作为第三方服务提供给租户的重要前提,因此,云租户隔离机制的安全有效性能否被租户信任,对云计算服务的推广非常关键.但在云计算这种第三方服务模式中,由于租户不能参与云服务基础设施及其安全隔离机制的建设和管理过程,因此他们难以对云租户隔离机制的安全有效性建立信心.将透明性要求视为可信云租户隔离机制的一部分,将云租户隔离机制和租户透明要求都转化为云服务系统中不同安全域之间的信息流,对云租户隔离机制进行定义,并制定云计算平台中的域间信息流策略控制方式,最后,基于信息流无干扰理论证明了所定义的云租户隔离机制在安全方面的有效性.

关键词: 云租户隔离;租户透明性;信息流策略;可信云;无干扰理论

中图法分类号: TP316

中文引用格式: 石勇,郭煜,刘吉强,韩臻,马威,常亮.一种透明的可信云租户隔离机制研究.软件学报,2016,27(6):1538−1548. http://www.jos.org.cn/1000-9825/4997.htm

英文引用格式: Shi Y, Guo Y, Liu JQ, Han Z, Ma W, Chang L. Trusted cloud tenant separation mechanism supporting transparency. Ruan Jian Xue Bao/Journal of Software, 2016,27(6):1538−1548 (in Chinese). http://www.jos.org.cn/1000-9825/4997.htm

Trusted Cloud Tenant Separation Mechanism Supporting Transparency

SHI Yong[1], GUO Yu[1], LIU Ji-Qiang[1], HAN Zhen[1], MA Wei[1], CHANG Liang[2]

[1](School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

[2](Guangxi Key Laboratory of Trusted Software (Guilin University of Electronic Technology), Guilin 541004, China)

Abstract: Tenant separation is a provision for cloud computing to be provided to tenants as a third party service, therefore the tenants' confidence in the security effectiveness of cloud tenant is critical to the promotion of cloud services. However, in a third party service such as cloud computing, tenants have few opportunities to take part in the construction and management of the infrastructure of cloud computing, making it hard for the tenants to trust the tenant separation mechanism in cloud. This paper views the transparency requirement as a part of trusted cloud tenant separation mechanism, implements a cloud tenant separation mechanism and its transparency requirement based on the inter-domain information flow control policy in cloud computing systems, and proves that the resulting cloud tenant separation mechanism is secure and effective by non-interference theory.

Key words: tenant separation; tenant transparency; information flow policy; trusted cloud; non-interference theory

云计算是一种由云计算服务商(cloud service provider,简称 CSP)面向多租户(tenant)提供的资源复用型服务[1,2],由于多个租户共享计算资源,因此,云租户隔离是否有效,是云计算服务能被租户接受的前提之一.所谓租

* 基金项目: 国家自然科学基金(61363030); 广西可信软件重点实验室研究课题(KX201531)

Foundation item: National Natural Science Foundation of China (61363030); Guangxi Key Laboratory of Trusted Software (KX201531)

收稿时间: 2015-08-14; 修改时间: 2015-10-09; 采用时间: 2015-12-05; jos 在线出版时间: 2016-01-21

CNKI 网络优先出版: 2016-01-22 11:01:40, http://www.cnki.net/kcms/detail/11.2560.TP.20160122.1101.012.html

户隔离是指在云计算平台中要禁止租户安全域之间的任何信息流动,以保证任何租户在云计算平台中的业务和数据不会受到其他租户的干扰或被其他租户观测到[3,4].针对云计算服务的租户隔离,目前有很多措施,比如网络隔离机制、虚拟机隔离技术、访问控制、安全审计、安全监控、存储和通信加密等等[1,5−7].

租户是否愿意采用云计算服务,一个很重要的因素是他们是否对云计算的租户隔离机制拥有足够信心.尽管前面提到的各种租户安全隔离措施在某种程度上可以提升租户的信心,但是仅仅依靠这些措施还不足以让租户充分相信云计算系统中租户隔离机制的有效性.这是因为租户没有机会参与云计算服务基础设施的建设,也不能参与云计算系统的运维管理.如果能够建立一种面向租户的透明性机制,使租户能够了解云计算租户隔离机制的原理、实现和运行细节,那么他们或许更愿意相信云计算系统中的租户隔离机制[8−14],因此,很多研究都集中在如何通过透明性机制实现可信的云服务.

但是这些基于透明性机制的可信云服务研究都主要偏重于对云服务某些属性的评估、测量、口碑和验证,从而建立一种信任感觉.这些属性可能是云服务的某些功能或性能,但是对相关属性的评估、测量、口碑和验证本质上是一种黑盒子的测量和验证,没有涉及到支撑云服务功能和性能的内部机制原理和细节,尤其是云租户隔离机制的内部策略细节,因此不能满足租户对云租户隔离机制的安全保证要求.

本文从云租户隔离机制的内部结构透明性出发,为租户提供云租户隔离机制内部策略细节的测量和验证方法,以实现高安全性的云租户隔离机制.本文认为:保证云租户隔离机制透明性的目的是让租户获得关于云计算租户隔离机制足够的信息,其本质上就是一种安全域之间的信息流动.即:要求云租户隔离机制的相关策略和运行过程信息从云计算平台安全域流向租户安全域,使租户可以随时对其状态进行测量,并验证其有效性.

为达到上述研究目标,本文以云计算系统中的域间信息流为基础,对云计算租户隔离机制和租户透明性要求展开研究,并在域间信息流策略层面将租户透明性要求和云租户隔离要求融为一体,建立面向透明性要求的云租户隔离机制;不仅如此,本文还运用信息流无干扰理论,对所提出的透明云租户隔离策略机制的安全有效性进行了证明.

1 相关研究

理论上,实现租户隔离有 3 种基本方法:空间隔离、时间隔离和密码隔离[15].作为云计算的基础技术,虚拟机技术既采用了空间隔离机制,也采用了时间隔离机制.比如对共享的 CPU 资源,虚拟机技术利用了时间隔离机制,它在不同的时刻将 CPU 计算资源分配给不同的虚拟机或租户.在虚拟机或租户环境切换时,它保存当前 CPU 状态,并恢复下一个被调度的虚拟机或租户环境.通过这种时间片分配调度机制,不同虚拟机或租户不能观测到其他虚拟机和租户的计算状态.对于磁盘等存储资源,虚拟机技术一般会采用空间隔离机制,将不同租户的数据隔离开[16,17].密码隔离也是云计算中经常用到的租户隔离方法,比如对于共享的网络资源,云计算一般会采用虚拟专用网络(virtual private network,简称 VPN)对不同租户的网络通信加以安全隔离.为了进一步加强 CPU 资源中的租户隔离强度,带密计算的方法也已被提出.所谓带密计算,是指租户的数据在被调度到 CPU 之前,无需像传统过程那样先被解密然后再计算,而是以密文的状态进入 CPU 并被计算和处理[18,19].

对云租户隔离机制的可信保证可以从 3 个方面开展研究:(1) 基于可信计算平台技术;(2) 基于软件结构和代码规模;(3) 基于租户透明要求和租户可控要求.

- 基于可信计算平台技术的可信保证机制以硬件密码模块(TPM)及在其上构建的可信软件栈(TSS)、可信网络连接(TNC)等为基础,保护云租户隔离机制的完整性.在这种方案中,云计算系统从一个可信的初始状态开始,通过信任链、可信证明、可信存储和可信网络等可信保证机制,确保在整个服务过程中的运行状态符合预期[20−26].为了使这种方案更加适用于云计算的要求,可信计算组织(TCG)还推出了动态可信根概念[26,27].基于可信计算平台技术的云租户隔离机制可信保证没有将租户作为可信保证的对象,它由 CSP 实现,它所考虑的和要满足的是 CSP 对云计算系统整体运行的单方面可信保证要求,而不仅仅是保证云租户隔离机制的有效性,其结果与租户对云租户隔离机制的可信保证要求没有任何直接关系.

- 软件的复杂度是影响软件可信度的重要因子,软件复杂度包括结构上的复杂度和软件代码的规模.一般认为:软件代码规模越大,代码的缺陷和安全漏洞就越多,其可信度也就越低.因此,尽量减少代码规模是提高云计算可信度的一种重要方法[28].但是对于云计算这种集多种组件于一体的综合服务平台,代码规模不可能无限制减小,因此,从结构上降低软件的复杂度就成为一种有意义的研究方向. Murray 认为,软件接口的数量和代码是很多重大软件错误的主要来源.他在文献[29]中提出要减少软件代码中的接口数量,以提高虚拟机监控器(virtual machine monitor,简称 VMM)等软件的质量和可信度.与基于可信计算平台技术的云租户隔离机制可信保证方法一样,基于软件结构和代码规模的云租户隔离机制可信保证方法也与租户没有关系,它只是 CSP 用来提高云服务系统安全运行的单方面的可信保证.
- 基于租户透明要求和可控要求的云租户隔离保证机制克服了前面两种方法的局限性,它以租户的可信要求为目标,真正提高租户对云租户隔离机制的信心和信任度.云计算服务是一种第三方服务机制,即,系统的建设和管理一般都是由 CSP 承担.要提高租户对云服务的信心,就必须要让租户实际参与云服务的管理[8,30],并让租户尽可能多的了解云租户隔离机制的内部策略和运行细节.比如: Kaufman 建议在云计算系统中为租户提供安全应用程序编程接口(application programming interface,简称 API),使租户可以自己对云计算服务过程进行监视和评估[8];Pauley 还给出了一个评分标准,帮助租户对云计算服务的透明程度进行有效评估[9];其他一些研究也给出了在云计算系统中如何提高透明性的方法和建议[12−14].但是,这些研究往往注重对云服务的某些外部属性进行测量(包括自我评估或口碑相传)和验证,比如云服务的某些功能和性能等.由于这些测量方法没有涉及到云租户隔离机制的内部结构和策略细节,因此它们很难获得云租户隔离机制的真实结构和运行状态,不能满足租户对云租户隔离机制的高安全性要求.

不同于上述研究,本文把透明性要求看做是一种云计算系统中不同安全域之间的信息流,它将云租户隔离机制的内部策略和实时运行信息从云管理平台安全域传送到租户安全域,从而为租户测量和验证云租户隔离机制提供了一种方法和手段;同时,由于这种测量和验证深入到了云隔离机制的内部原理和实时状态,因而它为租户确定云租户隔离机制是否可信提供了更高的信心保证.本文的主要贡献是从信息流策略机制出发,以租户透明要求为目标,将云租户隔离机制和租户透明要求都转化为云服务系统中不同安全域之间的信息流,并将二者实现融合,提出一种面向透明性要求的可信云租户隔离机制;不仅如此,本文还通过信息流无干扰理论证明了所提出机制的安全有效性,进一步提高了租户对云租户隔离机制的信心水平,这是本文的另一个主要贡献.

2 租户隔离策略机制

如果两个不同的安全域之间存在信息交换,那么它们之间必然存在共同的可访问地址空间或者通信连接.因此,要满足云计算平台中租户之间的安全隔离要求,必须要保证不同租户之间不存在交叉重叠的可访问地址空间,并且在不同租户之间不能存在直接的通信连接.本节基于云计算的资源复用要求和资源管理特征,提出同时满足云资源利用率最大化要求和安全隔离要求的云计算域间信息流策略机制.

2.1 云计算安全域划分

云计算系统中,计算资源包括计算时间资源和计算空间资源两个部分:

- 计算时间资源可以简单地用 CPU 计算时间来标识,包括总计算时间和单位时间内的 CPU 计算时间.云管理平台(cloud management platform,简称 CMP)根据云计算的服务等级协定(service level agreement,简称 SLA)为租户分配相应的 CPU 计算时间;
- 计算空间资源包括内存、磁盘、I/O 等物理和逻辑存储资源,其范围可用资源所在的地址空间来标识.

本文为了简化问题讨论,对计算时间资源不加考虑,只采用计算资源地址空间来表示云计算资源.这样,系统关于计算资源的管理就表现为对资源地址空间的管理.比如:系统为租户新分配一个虚拟机,就意味着该租户所拥有的计算资源地址空间增加;系统或租户对计算资源的操作,表现为对资源地址空间的内容读写.

在云计算系统中,云计算平台由多个安全域组成,它包括:

1) CMP:CMP 与租户通信并为租户提供服务;
2) 租户(tenant)域:它们是由 CMP 根据服务合约分配给相应租户的;
3) 系统可分配资源域(system resource pool,简称 SRP):这一类资源是由 CMP 管理,但是可能会根据需要分配给租户使用.

任何状态下,这 3 类资源是云计算系统地址空间的一个划分,相互之间没有重叠.

云计算的这种地址空间划分体现了云计算的安全隔离特征,但是这种划分是动态变化的.系统通过 CMP,动态从 SRP 中为租户分配资源,或将租户域中的资源进行回收,归还到系统可分配资源域中.

2.2 租户隔离

本文用 $M(D,\rightarrowtail)$表示云计算系统:

- $D=\{P,R,T_1,T_2,\ldots,T_n\}$,其中,$P$ 表示 CMP 所在安全域,R 表示 SRP,$T_i(1\leqslant i\leqslant n)$表示租户 i 对应的安全域;
- $\rightarrowtail\subseteq D\times D$,对$\forall u,v\in D,u\rightarrowtail v$ 表示信息可以从安全域 u 流向安全域 v,或 u 对 v 有干扰.

显然,$\rightarrowtail$满足自反关系.为方便起见,符号“$\nrightarrow$”表示无干扰,$u\nrightarrow v$ 表示 u 对 v 没有干扰.

用 H 表示 $M(D,\rightarrowtail)$的地址空间的集合,S 表示系统 $M(D,\rightarrowtail)$的状态集,$s_0\in S$ 表示系统初始状态.根据第 2.1 节,在任何状态下,$P,R,T_1,T_2,\ldots,T_n$ 都是 H 的一个划分,即有 $H=P\cup R\cup T_1\cup\ldots\cup T_n$.用函数 $h:S\times D\rightarrow 2^H$ 表示特定系统状态下安全域对应的实际地址空间,函数 $domh:S\times H\rightarrow D$ 表示某个地址空间在特定状态下所属的安全域;V 表示地址空间 H 的全部可能取值集合,函数 $val:S\times H\rightarrow V$ 表示 $M(D,\rightarrowtail)$中某个地址在特定状态下的值;为简单起见,我们用赋值“0”的方式表示使某个地址(或设备)复位或清空,比如,$val(s,h)=0$ 表示在状态 s 下使地址 h 复位或清空.

用 A 表示系统的全部动作集合,O 是系统输出集合,函数 $dom:A\rightarrow D$ 表示每个动作所对应的安全域,$step:S\times A\rightarrow S$ 是系统状态转换函数,$obs:S\times D\rightarrow O$ 表示特定安全域在某个状态下所观察到的系统输出.$s\cdot\alpha$表示从状态 s 经过动作序列$\alpha\in A^*$所到达的状态;如果用ε表示空动作序列,$a\in A$,那么 $s\cdot\varepsilon=s,s\cdot\alpha a=step(s\cdot\alpha,a)$.

特定地址空间对应的值与系统状态相关,它们可能由于系统中的动作发生改变.不失一般性,我们假定$\forall s,t\in S,r\in H,a\in A,val(s,r)=val(t,r)\Rightarrow val(step(s,a),r)=val(step(t,a),r)$,即:对于一个具体的存储地址,其存储值的变化只与系统动作相关.

某个安全域中所观察到的系统输出包括两个内容:地址空间范围及其每个地址对应的值,即,系统输出函数可以具体定义如下:对$\forall d\in D,s\in S,obs(s,d)=\{(m,val(s,m))|\forall m\in h(s,d)\}$.

2.2.1 通道(channel)

由于 $D=\{P,R,T_1,T_2,\ldots,T_n\}$是 H 的一个划分,云计算系统 $M(D,\rightarrowtail)$中任何两个不同安全域之间不存在共同的可访问地址空间,它们之间只能通过通道(channel)来实现域间通信.根据云计算平台的域间隔离要求,任何两个租户之间都不应该有信息交换,但是为了实现云资源的动态复用目标,每个租户都应该能与 CMP 进行通信,以提交资源申请或向系统退还不用的资源.为避免资源的滥用,云计算系统中禁止租户直接访问 SRP,租户只能通过 CMP 来获取或退还资源.

为简化对通道的描述,本文假定一个通道只支持单向通信.用 $C\subseteq H\times H\times S$ 表示 $M(D,\rightarrowtail)$在特定状态下的通道集合,对于 $c=\langle h_1,h_2,s\rangle\in C,h_1$ 表示通道 c 的源地址,h_2 表示通道 c 的目的地址.用 $src:C\rightarrow D$ 表示通道的源发域,即,写通道的安全域;$tgt:C\rightarrow D$ 表示通道的目的域,即,读取通道的安全域.支持单向通信要求:

$$\forall c\in C\Rightarrow src(c)\cap tgt(c)=\varnothing.$$

为了满足租户隔离要求,通道要么源发于 CMP,要么终结于 CMP,即,$\forall c\in C\Rightarrow src(c)=P\vee tgt(c)=P$.

同时,所有的租户安全域必须在 CMP 管理之下,即:

$$\forall u\in D-P\Rightarrow\exists c_1=\langle h_1,h_2,s\rangle\in C\wedge domh(s,h_1)=u\wedge domh(s,h_2)=P\wedge\exists c_2=\langle h_3,h_4,s\rangle\in C\wedge domh(s,h_3)=P\wedge domh(s,h_4)=u.$$

2.2.2 资源重用和剩余信息保护

第 2.2.1 节中提出的通道及其规则可以禁止租户安全域之间的显式信息流,但是云计算的资源复用机制还是可能导致租户安全域之间的隐式信息流.比如:如果某个租户归还给系统的存储资源没有被清理干净就分配给下一个租户的话,那么残留在这些存储资源上的信息就会被其他租户观察到.

为了消除资源重用机制下租户安全域之间的这种隐式信息流,系统需要满足以下资源管理要求:

- 要求 1:对$\forall r\in H$,有 $domh(s_0,r)=R\Rightarrow val(s_0,r)=0$;
- 要求 2:对$\forall s\in S,r\in H,a\in A$,有 $domh(step(s,a),r)\neq domh(s,r)\wedge domh(s,r)=R\Rightarrow dom(a)=P$;
- 要求 3:对$\forall s\in S,r\in H,a\in A$,有:

  $domh(step(s,a),r)\neq domh(s,r)\wedge domh(s,r)\neq R\Rightarrow domh(step(s,a),r)=R\wedge val(step(s,a),r)=0\wedge dom(a)=P$.

要求 1 指出:系统初始化时,SRP 中的所有地址空间必须被清空;要求 2 指出:所有资源必须由 CMP 从 SRP 中调取并分配给租户;要求 3 指出:云计算中的资源要么继续由租户保留使用,要么被 CMP 收回,并在清空后归还到 SRP 中.

2.2.3 租户透明机制

租户透明机制意味着在不违背租户隔离机制的条件下,CMP 中的状态信息应该尽可能地对租户透明.CMP 中的状态信息可以划分为 3 类:第 1 类状态信息与所有租户的隐私保护密切相关,不能对任何租户开放,一旦开放就会让租户了解到其他租户的信息;第 2 类状态信息与具体租户隐私无关,可以对所有租户开放,比如云计算基础设施中所用到基础软件的版本信息等;第 3 类状态信息与特定的租户相关,只能开放给相应租户.

用 $P_t=\{P_{nr},P_r,T_{1r},T_{2r},\ldots,T_{nr}\}$表示 P 的一个划分,其中,

- P_{nr} 表示第 1 类状态信息,不能对任何租户开放;
- P_r 表示第 2 类状态信息,可以对所有租户开放读取功能,但是任何租户都不能改变;
- $T_{ir}(1\leqslant i\leqslant n)$表示第 3 类状态信息,即:只对租户 i 开放,租户 i 可以读取或改变其状态.

可以采用通道来实现租户透明机制,比如,用一个源发于 CMP 的通道来向租户提供他们希望知道并且允许知道的系统状态信息.

用函数 $b:S\times H\rightarrow P_t$ 来表示 CMP 中的一个地址空间隶属于 P_t 中的一个区域,我们有以下规则:

规则 1. $\forall c=\langle h_1,h_2,s\rangle\in C\Rightarrow b(s,h_1)\notin P_{nr}$.

规则 2. $\forall c=\langle h_1,h_2,s\rangle\in C\wedge b(s,h_1)\in T_{ir}\Rightarrow domh(s,h_2)\in T_i$.

规则 3. $\forall c=\langle h_1,h_2,s\rangle\in C\wedge domh(s,h_2)\in T_i\Rightarrow b(s,h_1)\in T_{ir}\cup P_r$.

规则 4. $\forall c=\langle h_1,h_2,s\rangle\in C\wedge domh(s,h_1)\in T_i\Rightarrow b(s,h_2)\in T_{ir}$.

规则 5. $\forall i,j,1\leqslant i\leqslant n,1\leqslant i\leqslant n,i\neq j\Rightarrow T_{ir}\cap T_{jr}=\varnothing$.

规则 1 表示不可能有一个通道源发于 CMP 中不能对租户开放的区域;规则 2 表示一个通道如果源发于 CMP 中只对具体租户开放的区域,那么它只能终止于该租户安全域;规则 3 表示如果一个通道终止于某个租户,那么它要么源发于 CMP 中的对所有租户开放的区域,要么源发于只对该租户开放的区域;规则 4 表示如果一个通道源发于一个租户安全域,那么它一定终止于 CMP 中只对特定租户开放的区域.规则 5 表示 CMP 中对不同租户开放的区域没有交集.

3 可行性分析与验证

3.1 可行性分析

上述云租户隔离策略机制在技术上是可行的和合理的.

首先是租户安全域的资源隔离机制,其主要难点在于在共享平台上实现不同租户安全域占用资源之间的安全隔离,比如分配给租户的虚拟机组、存储资源以及网络资源与其他租户之间没有重叠交叉部分.由于虚拟机技术只是实现虚拟机之间的隔离,而每个租户可能同时拥有多个虚拟机,因此不同租户的虚拟机组之间则需要虚拟网络(VLAN)等技术来实现标识和隔离,比如 802.1Q;对于租户存储资源的隔离,则可能需要在存储系统中通过访问控制和数据加密等安全机制来实现;在云计算平台的共享网络中,要实现不同租户的区隔,VPN 是一种可选的机制;

其次是剩余信息保护机制的可行性问题.在云计算资源被重新分配时,第 2.2.2 节通过对回收资源进行清空设置来避免租户域之间可能存在的间接信息流,从而实现剩余信息保护机制.对于计算资源,不同的云计算服务

类型面临着不同的实现难度,比如基础设施即服务(infrastructure as a service,简称 IaaS)和平台即服务(platform as a service,简称 PaaS),租户退还虚拟机后,CSP 可以通过删除、重新创建或克隆模式实现计算资源的清空;但是对软件即服务(software as a service,简称 SaaS),租户退还计算资源后,CSP 对相关资源的清空就比较困难,其原因是这些租户在资源使用过程中,可能对底层系统平台的状态产生影响,这些影响难以通过系统重启来清除,因为可能还有其他租户在使用这些平台,因此需要在 SaaS 的服务相关应用层面提供支持,在租户退还服务资源后,对相关状态进行清理或清空;

再次是剩余信息保护机制的性能问题.系统在清空被租户退还的磁盘等存储资源过程中,对磁盘的清空会涉及到对退还磁盘空间的重写(否则,前租户的相关信息还会保存在磁盘上),普通磁盘(如 SATA 和 SAS 磁盘)的写过程极其耗时;在面向大量租户服务的云计算系统中,磁盘资源的这种动态复用机制将导致大量的磁盘重写行为,而磁盘的 IOPS(每秒读写次数)是影响云计算系统整体性能的一个主要因素.要解决这一性能问题,可以采用磁盘异步清空方式,所谓磁盘异步清空方式是指:在磁盘存储资源被退还后,系统将这部分磁盘空间标记为“未清空”状态,所有状态为“未清空”的磁盘存储资源不能被重新分配给租户;系统通过一个专门的异步进程来处理“未清空”的磁盘存储空间,在不影响云计算服务整体性能的前提下,该异步进程通过利用系统空闲时间片对相应磁盘空间进行重写;只有清空后磁盘空间才可以被重新分配给租户;要保证磁盘异步清空方式能够正常工作,CSP 需要配置一定容量的冗余磁盘空间;

最后是通道以及相应的租户透明机制.通道作为安全域和 CMP 之间的通信载体,既需要承担一定管理命令的传输,如 Hypervisor 向虚拟机发送的管理命令,也需要承担安全域和 CMP 之间的数据传递.对于前者而言,往往体现在虚拟系统内部,如 Xen 中的 Event Channel 和 Hypercall 等;而对于后者而言,主要考虑的应当是数据传输中的机密性和完整性,因此,VPN 是一种良好的应对方法.在通道能够保障传输信息的机密性和完整性的基础上,透明机制的可行性问题则主要集中在对 CMP 中相关信息如租户虚拟机运行状态和当前执行的安全策略的封装上,应当能够确保这些信息的可靠性和可验证性.针对这一问题,基于可信计算和虚拟化技术相结合的 vTPM 将会是一种可行的解决方案,CMP 从不同的虚拟机中收集当前的虚拟机运行状态信息,经过虚拟机的 vTPM 中的 AIK 签名后,汇总至 CMP 进行验证、封装并使用 CMP 的 AIK 再次签名后,经由通道发送给租户,确保了这些透明性信息的可靠.

3.2 原型系统验证

图 1 是原型系统示意图,验证了本文提出的若干关键技术.

在图 1 中,为租户提供服务的云计算环境包含 3 个部分:CMP、计算集群和存储集群,其中,计算集群主要承担虚拟机即安全域的运行,而存储集群则主要为虚拟机提供存储服务.本文提出的隔离机制在原型系统中主要体现在如下几个方面:

1. 为不同租户提供服务的虚拟机(组)之间采用 VLAN 技术进行了隔离,满足了租户安全域的隔离机制;
2. 使用访问控制技术确保不同虚拟机(组)之间对物理存储的访问安全受控,实现了对存储资源的隔离;
3. 租户通过 VPN 连接到 CMP 的对外接口来使用云计算服务,是一种对租户空间的隔离;
4. 而在存储集群中,基于母本和 COW(copy-on-wirte)机制,使得存储集群能够实现磁盘资源的初始分配、回收和再分配,同时也考虑到了磁盘的异步清空,实现了对剩余信息的保护;
5. 云环境中的通道则体现在租户与 CMP 之间的 VPN 连接和 CMP 通过调用 Event Channel 等对虚拟机进行管理等两个方面;
6. CMP 从虚拟机中收集当前的透明性证明,封装后与云计算服务一同提供给租户,则是原型系统对透明性要求的体现.

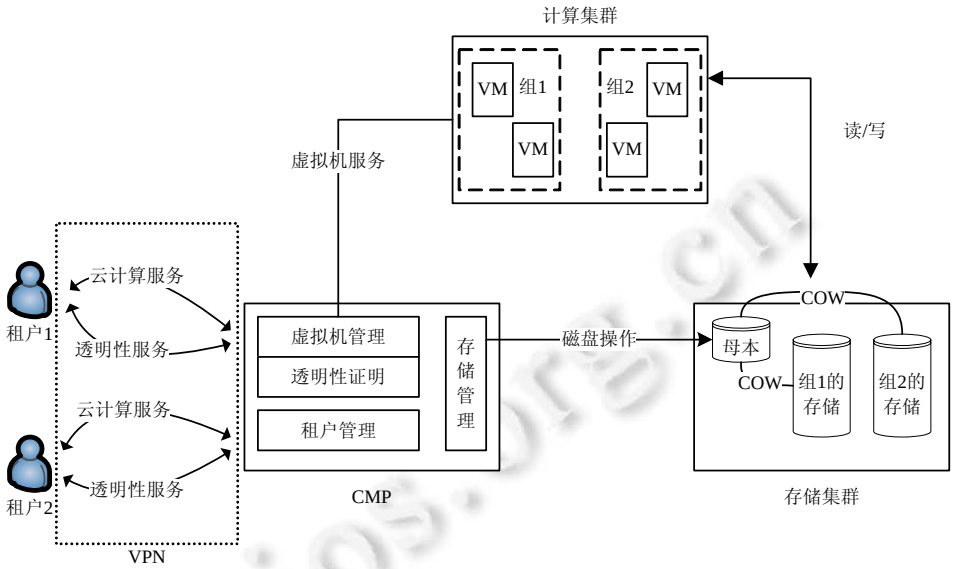

Fig.1 Schematic diagram of prototype system

图 1 原型系统示意图

4 安全性分析

要进一步提高租户对云租户隔离机制的信心,CSP 还应该按照 TCSEC[31]和 CC[32]的要求,为云租户隔离机制建立相关的形式化策略模型,并提供相关的有效性证据.云租户隔离要求本质上是对租户安全域之间的信息流控制机制,即,要限制各租户安全域之间直接和间接的信息流.要证明云计算各安全域间信息流策略机制对租户的安全隔离和保护能力,无干扰理论(non-interference theory)无疑是一种很好方法的理论工具和方法.但是在云计算服务中,采用无干扰理论工具和方法证明租户隔离机制的安全有效性时,要充分考虑并基于云计算服务的具体特征,比如资源的动态性复用能力.

无干扰概念在信息安全领域中用来描述安全域之间的干扰关系,即,不同安全域之间的信息流方式.如果安全域 u 的任何动作都不会让安全域 v 感知到,即,u 的这些动作不会改变 v 所能观察到的系统输出结果,那么就表示 u 对 v 无干扰.Goguen 等人在 1982 年首次提出了无干扰理论[33],并提出了用来证明信息流策略模型的安全性的展开定理(unwinding theorem)[34].可以说,Goguen 的研究奠定了无干扰理论的基础.但是其结论也存在相当的局限性,它们只能应用于具有传递性质的安全策略环境中,比如安全策略允许信息可以从安全域 A 流向域 B,同时也允许从域 B 流向域 C,那么系统一定要允许信息能从 A 流向 C.这一局限性严重限制了它在实际环境中的应用,在很多类似场景中,需要 B 作为 A 和 C 之间的信息流安全控制或审查者,A 到 C 的信息流必须经过 B,但是不能直接由 A 流向 C.在这类场景中,Goguen 等人的研究结论显然不再适用.针对这一局限性,Haigh 等人提出了非传递性安全策略环境下的非传递无干扰理论[35],并获得了普遍认可.但是 Rushby 也发现了他们在相关定义中存在的问题,表示文献[35]中的“SAT MDS 展开定理”结论并不正确,同时对其中的非传递无干扰定义进行了修改,并给出了相关的非传递性策略展开定理(unwinding theorem for intransitive policies)[36].

但是 Rushby 对非传递无干扰的定义过于严格,他对动作序列的严格性并不完全符合域间干扰的真实含义.Meyden 发现了这一问题,并纠正了 Rushby 的相关定义,提出了 TA-安全和 TO-安全两个新的非传递无干扰模型[37].

无干扰理论已成功地应用于多个关键项目之中,比如美军的 F35 战斗机、DDG 1000 驱逐舰以及美国陆军的未来作战系统等军工项目中[38].

下面我们将证明第 2 节所给出的租户隔离策略机制是安全有效的.在给出具体的证明之前,先介绍 Meyden

的 TA-安全判定定理[37].

对于 $M(D,\rightarrowtail),u\in D,a\in A,\alpha\in A^*$,函数 ta_u 定义如下:

1. $ta_u(\varepsilon)=\varepsilon$;
2. 如果 $dom(a)\nrightarrow u$,那么 $ta_u(\alpha a)=ta_u(\alpha)$;
3. 如果 $dom(a)\rightarrowtail u$,那么 $ta_u(\alpha a)=(ta_u(\alpha),ta_{dom(a)}(\alpha),a)$.

以此为基础,Meyden 给出了系统安全定义:系统 $M(D,\rightarrowtail)$中,对$\forall u\in D,\forall\alpha\in A^*$和$\alpha'\in A^*$,如果 $ta_u(\alpha)=ta_u(\alpha')$,都有 $obs(u,s_0\cdot\alpha)=obs(u,s_0\cdot\alpha')$,那么系统 $M(D,\rightarrowtail)$对策略$\rightarrowtail$是 TA-安全的.

Meyden 还给出了以下系统安全判定定理[37]:

定理 1. 如果系统 $M(D,\rightarrowtail)$存在关于策略$\rightarrowtail$的弱展开(weak unwinding),则 $M(D,\rightarrowtail)$关于策略$\rightarrowtail$是 TA-安全的.

其中,系统 $M(D,\rightarrowtail)$关于策略$\rightarrowtail$的弱展开是指满足以下条件的关于 D 的关系族$\sim_u$:

1. 如果 $s\sim_u t$,那么 $obs(u,s)=obs(u,t)$; (输出一致性,简称 OC)
2. 如果 $s\sim_u t$,并且 $s\sim_{dom(a)}t$,那么 $s\cdot a\sim_u t\cdot a$; (弱单步一致性,简称 WSC)
3. 如果 $dom(a)\nrightarrow u$,那么 $s\sim_u s\cdot a$. (局部符合性,简称 LR)

如非特别说明,本文以后提到的系统 $M(D,\rightarrowtail)$都是指满足第 2 节各种定义、要求和规则的云计算系统.

首先我们给出云计算系统 $M(D,\rightarrowtail)$的域间信息流相关定义.

定义 1. 云计算系统 $M(D,\rightarrowtail)$中的域间信息流集合定义如下:

1. $F=\varnothing$;
2. 如果$\forall c=\langle h_1,h_2,s\rangle\in C$,那么 $F=F\cup\{|\langle h_1,h_2,s\rangle|\}$;
3. 如果 $f_1=|\langle h_1,h_2,s\rangle|,f_2=|\langle h_2,h_3,s\rangle|\in F$,那么 $F=F\cup\{|\langle h_1,h_3,s\rangle|\}$.

其中,$f=|\langle h_1,h_2,s\rangle|\in F$,$h_1$ 表示信息流 f 的源地址,h_2 表示信息流 f 的目的地址.

定义 2. 云计算系统 $M(D,\rightarrowtail)$中,域间干扰关系$\rightarrowtail$定义为:

$$\forall u,v\in D,u\rightarrowtail v \text{ 当且仅当}(\exists f)(f=|\langle h_1,h_2,s\rangle|\in F\wedge domh(s,h_1)=u\wedge domh(s,h_2)=v).$$

引理 1. 云计算系统 $M(D,\rightarrowtail)$中,对于$\forall s\in S,r\in H$,如果 $domh(s,r)=R$,那么 $val(s,r)=0$.

证明:由第 2.2.2 节中的要求 1~要求 3,通过递归法可以证明(证明过程略). □

引理 1 表示任何状态下系统可分配资源中的地址空间都是清空状态,这避免了一个租户从新分配资源中获取其他租户信息的可能.

引理 2. 云计算系统 $M(D,\rightarrowtail)$中,对于$\forall a\in A,u\in D-P$,如果 $dom(a)\nrightarrow u$,那么 $dom(a)\neq P$.

证明:根据第 2.2.1 节中的假设,所有租户安全域必须受 CMP 管理,即:

$$\forall u\in D-P\Rightarrow\exists c_1=\langle h_1,h_2,s\rangle\in C\wedge domh(s,h_1)=P\wedge domh(s,h_2)=u.$$

因此,如果 $dom(a)\nrightarrow u$,那么必然有 $dom(a)\neq P$. □

引理 3. 云计算系统 $M(D,\rightarrowtail)$中,对于$\forall u,v\in D-P$,有 $u\rightarrowtail v\Rightarrow u=v$.

证明:由 $u,v\in D-P,u\rightarrowtail v$,根据定义 2,$(\exists f)(f=|\langle h_1,h_2,s\rangle|\in F\wedge domh(s,h_1)=u\wedge domh(s,h_2)=v)$.假设 $u\neq v$,u,v 之间至少存在一条通道,又根据第 2.2.1 节通道性质$\forall c\in C\Rightarrow src(c)=P\vee tgt(c)=P$,$u,v$ 必须经过 P 传递信息,因此存在信息流:

$$u\rightarrowtail P\rightarrowtail\ldots\rightarrowtail P\rightarrowtail v.$$

先考虑最简单情况,$u\rightarrowtail P\rightarrowtail v$.由 $u\rightarrowtail P$,有 $\exists c_1=\langle h_1,h_1',s\rangle\in C\wedge domh(s,h_1)=u\wedge domh(s,h_1')=P$,根据第 2.2.3 节的规则 4,有 $b(s,h_1')\in T_{ur}$;由 $P\rightarrowtail v$,有 $\exists c_2=\langle h_2',h_2,s\rangle\in C\wedge domh(s,h_2')=P\wedge domh(s,h_2)=v$,根据第 2.2.3 节的规则 3,$b(s,h_2')\in T_{vr}\cup P_r$.根据第 2.2.3 节的规则 5,因为 $u\neq v$,有 $T_{ur}\cap(T_{vr}\cup P_r)=\phi$,故 $b(s,h_1')\cap b(s,h_2')=\varnothing$,与 $u\rightarrowtail P\rightarrowtail v$ 矛盾,所以 $u=v$.

递归可推出 $u\rightarrowtail P\rightarrowtail t_1\rightarrowtail P\rightarrowtail t_2\rightarrowtail P\rightarrowtail\ldots\rightarrowtail t_n\rightarrowtail P\rightarrowtail v$,$t_i\in D-P$,$1\leqslant i\leqslant n$ 时,$u=t_1=t_2=\ldots=t_n=v$. □

引理 3 表示任何状态下,任意两个租户安全域之间不可能通过通道传递信息.

引理 4. 云计算系统 $M(D,\rightarrowtail)$中,对于$\forall u,v\in D$,有 $u\neq v,u\rightarrowtail v\Rightarrow u=P\vee v=P$.

证明:假设 $u\neq P\wedge v\neq P$,由于 $u\rightarrowtail v$,根据引理 3,有 $u=v$,与 $u\neq v$ 矛盾.故假设不成立. □

引理 5. 云计算系统 $M(D,\rightarrowtail)$中,对于$\forall a\in A,\forall s,t\in S,u\in D$,有:

$$obs(s,u)=obs(t,u)\wedge obs(s,dom(a))=obs(t,dom(a))\Rightarrow obs(step(s,a),u)=obs(step(t,a),u).$$

证明:根据第 2.2 节关于函数 $obs(\cdot)$的定义,$obs(\cdot)$由域地址空间的范围及其对应值两个因素确定,$obs(s,u)=obs(t,u)$表示在状态 s 和 t 下,安全域 u 的地址空间范围相同,并且每个地址对应的值也相同.

- 当 $dom(a)\nrightarrow u$ 时,即:动作 a 既不会改变 u 的地址空间范围,也不会改变每个地址对应的值,所以有:

$$obs(s,u)=obs(t,u)\Rightarrow obs(step(s,a),u)=obs(step(t,a),u);$$

- 当 $dom(a)\rightarrowtail u$ 时,根据引理 4,分 3 种情况:
  1. 如果 $dom(a)=u$,$dom(a)$为地址读写动作,a 将不会改变 u 的地址空间范围.按照第 2.2 节中的假定,对$\forall s,t\in S,r\in H,a\in A,val(s,r)=val(t,r)\Rightarrow val(step(s,a),r)=val(step(t,a),r)$,因此:

$$obs(step(s,a),u)=obs(step(t,a),u);$$

  2. 如果 $dom(a)\neq u$,$dom(a)=P$,$dom(a)$资源管理类.此时 $dom(a)=P$,且 a 为 u 分配资源或从 u 回收资源.此种情况下,a 将改变 u 的地址空间范围,但不会改变每个地址对应的值.按照引理 1,如果是新分配资源,其地址空间的值都为 0;如果是回收资源,那么余下的地址空间值不会发生变化.因此在状态 s 和 t 下,动作 a 完成后,u 的地址空间范围及每个地址对应的值也保持相同,即:

$$obs(step(s,a),u)=obs(step(t,a),u);$$

  3. 如果 $dom(a)\neq u$,$u=P$,$dom(a)$读写 $T_{dom(a)r}$,上报租户透明机制中的第三类信息.由于 $obs(s,dom(a))=obs(t,dom(a))$,所以在 s 和 t 状态下,动作 a 将 $dom(a)$中相同的状态信息上报到 $T_{dom(a)r}$,而不影响 P_t 中其他地址空间的值.故有 $obs(step(s,a),u)=obs(step(t,a),u)$. □

定理 2. 云计算系统 $M(D,\rightarrowtail)$关于策略"$\rightarrowtail$"是 TA-安全的.

证明:要证明 $M(D,\rightarrowtail)$关于策略"$\rightarrowtail$"是 TA-安全的,根据定理 1,必须要证明 $M(D,\rightarrowtail)$存在关于策略"$\rightarrowtail$"的弱展开满足 OC,WSC 和 LR 要求.

- 定义 $M(D,\rightarrowtail)$上关于 D 的关系族$\sim_u$为:$s\sim_u t$ 当且仅当 $obs(s,u)=obs(t,u)$.显然,OC 满足;
- 根据引理 5,显然有 $s\sim_u t\wedge s\sim_{dom(a)}t\Rightarrow step(s,a)\sim_u step(t,a)$,即,WSC 满足;
- 最后,我们需要证明 LR,即 $dom(a)\nrightarrow u\Rightarrow s\sim_u step(s,a)$.因为 $dom(a)\nrightarrow u$,由引理 2 知道 $dom(a)\neq P$,因此 $dom(a)$不会改变 u 的地址空间范围;又因为 $dom(a)\nrightarrow u$,因此 $dom(a)$不会改变 u 的地址空间所对应的值.综合上述分析,$obs(s,u)=obs(step(s,a),u)$.按照$\sim_u$的定义,有 $s\sim_u step(s,a)$,即,LR 满足. □

定理 2 证明了第 2 节中的云计算租户隔离策略机制能够保证不同租户之间的安全隔离.

5 结 论

对云租户隔离机制的可信保证研究包括多个层面,包括面向系统运行过程的完整性保证、租户隔离机制策略的形式化描述和证明、租户透明性和可控性保证、租户信任的评估及传递计算模型等多个方面.本文以租户透明性为基础,将租户透明要求视作云计算平台到租户安全域之间的一种信息流,并把这种信息流融合到租户隔离机制之中,作为租户隔离机制中的一部分策略规则,从而实现对透明性要求的形式化描述,并对相关租户隔离模型的有效性进行了证明.这种把抽象的系统可信要求转化为具体的形式化规则的方法是一种创新,为以后的类似研究提供了参考和借鉴.

References:

[1] Cloud Security Alliance. Security guidance for critical areas of focus in cloud computing. http://www.cloudsecurityalliance.org/guidance/csaguide.pdf

[2] NIST. The NIST definition of cloud computing. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[3] Survey: Cloud computing 'no hype', but fear of security and control slowing adoption. http://www.circleid.com/posts/20090226_cloud_computing_hype_security/

[4] F5 networks: Cloud computing survey results June—July 2009. 2009. http://www.f5.com/pdf/reports/cloud-computing-survey-results-2009.pdf

[5] Mather T, Kumaraswamy S, Latif S. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance. O'Reilly, 2009.

[6] Almond C. A Practical Guide to Cloud Computing Security. A White Paper from Accenture and Microsoft. 2009.

[7] Wang C, Wang Q, Ren K, Lou WJ. Privacy-Preserving public auditing for data storage security in cloud computing. In: Proc. of the 2010 IEEE INFOCOM. IEEE, 2010. 1–9. [doi: 10.1109/INFCOM.2010.5462173]

[8] Kaufman LM. Can a trusted environment provide security? Security & Privacy, IEEE, 2010,8(1):50–52. [doi: 10.1109/MSP.2010.33]

[9] Pauley WA. Cloud provider transparency: An empirical evaluation. Security & Privacy, IEEE, 2010,8(6):32–39. [doi: 10.1109/MSP.2010.140]

[10] Khan KM, Malluhi Q. Establishing trust in cloud computing. IT Professional, 2010,12(5):20–27. [doi: 10.1109/MITP.2010.128]

[11] Huang J, Nicol DM. Trust mechanisms for cloud computing. Journal of Cloud Computing, 2013,2(1):1–14. [doi: 10.1186/2192-113X-2-9]

[12] Sunyaev A, Schneider S. Cloud services certification. Communications of the ACM, 2013,56(2):33–36. [doi: 10.1145/2408776.2408789]

[13] Kumar N, Chakraborti B, Kumar A, Giri S. Reduction of cost by implementing transparency in cloud computing through different approaches. In: Proc. of the 2014 Int'l Conf. on Advanced Communication Control and Computing Technologies (ICACCCT). IEEE, 2014. 1723–1725. [doi: 10.1109/ICACCCT.2014.7019403]

[14] Dev H, Ali ME, Sen T, Basak M. AntiqueData: A proxy to maintain computational transparency in cloud. In: Proc. of the Database Systems for Advanced Applications. Berlin, Heidelberg: Springer-Verlag, 2014. 256–267. [doi: 10.1007/978-3-662-43984-5_19]

[15] Rushby J. A formal model for MILS integration. Project Report, Menlo Park, CA: Computer Science Laboratory, SRI International, 2008.

[16] Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A. Xen and the art of virtualization. ACM SIGOPS Operating Systems Review, 2003,37(5):164–177. [doi: 10.1145/1165389.945462]

[17] Mell P, Grance T. The NIST definition of cloud computing. National Institute of Standards and Technology, 2009,53(6):50.

[18] Chow R, Golle P, Jakobsson M, Masuoka R, Molina J. Controlling data in the cloud: Outsourcing computation without outsourcing control. In: Proc. of the 2009 ACM Workshop on Cloud Computing Security. ACM Press, 2009. 85–90. [doi: 10.1145/1655008.1655020]

[19] Khan KM, Malluhi Q. Establishing trust in cloud computing. IT Professional, 2010,12(5):20–27. [doi: 10.1109/MITP.2010.128]

[20] TCG. https://www.trustedcomputinggroup.org/home

[21] Patel A, Dansena P. TPM as a middleware for enterprise data security. Int'l Journal of Computer Science and Mobile Computing, 2013,2(7):327–332.

[22] Kekkonen T, Kanstrén T, Hatonen K. Towards trusted environment in cloud monitoring. In: Proc. of the 2014 11th Int'l Conf. on Information Technology: New Generations (ITNG). IEEE, 2014. 180–185. [doi: 10.1109/ITNG.2014.104]

[23] Li XY, Zhou LT, Shi Y, Guo Y. A trusted computing environment model in cloud architecture. In: Proc. of the 2010 Int'l Conf. on Machine Learning and Cybernetics (ICMLC). IEEE, 2010. 2843–2848. [doi: 10.1109/ICMLC.2010.5580769]

[24] Varadharajan V, Tupakula U. TREASURE: Trust enhanced security for cloud environments. In: Proc. of the 2012 IEEE 11th Int'l Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2012. 145–152. [doi: 10.1109/TrustCom.2012.283]

[25] Chen C, Raj H, Saroiu S, Wolman A. cTPM: A cloud TPM for cross-device trusted applications. In: Proc. of the 11th USENIX Conf. on Networked Systems Design and Implementation. 2014. 187–201.

[26] Gebhardt C, Dalton CI, Brown R. Preventing hypervisor-based rootkits with trusted execution technology. Network Security, 2008, 11(2008):7–12. [doi: 10.1016/S1353-4858(08)70128-4]

[27] Nie C. Dynamic root of trust in trusted computing. TKK T1105290 Seminar on Network Security, 2007.

[28] Singaravelu L, Pu C, Härtig H, Helmuth C. Reducing TCB complexity for security-sensitive applications: Three case studies. ACM SIGOPS Operating Systems Review, 2006,40(4):161−174. [doi: 10.1145/1217935.1217951]

[29] Murray DG, Milos G, Hand S. Improving Xen security through disaggregation. In: Proc. of the 4th ACM SIGPLAN/SIGOPS Int'l Conf. on Virtual Execution Environments. ACM Press, 2008. 151−160. [doi: 10.1145/1346256.1346278]

[30] Chen Y, Paxson V, Katz RH. What's new about cloud computing security. Berkeley Report, No.UCB/EECS-2010-5, University of California, 2010.

[31] Department of Defense. Trusted computer system evaluation criteria (orange book). DoD Computer Security Center, 1983. http://csrc.nist.gov/publications/history/dod85.pdf

[32] ISO/IEC 15408 Standard. Common criteria for information technology security evaluation version 3.1 revision 4. 2014. http://www.commoncriteriaportal.org/cc/

[33] Goguen JA, Meseguer J. Security policies and security models. In: Proc. of the IEEE Symp. on Security and Privacy. 1982. [doi: 10.1109/SP.1982.10014]

[34] Goguen JA, Meseguer J. Inference control and unwinding. In: Proc. of the 1984 Symp. on Security and Privacy. Oakland: IEEE Computer Society, 1984. 75−86. [doi: 10.1109/SP.1984.10019]

[35] Haigh JT, Young WD. Extending the noninterference version of MLS for SAT. IEEE Trans. on Software Engineering, 1987,2: 141−150. [doi: 10.1109/TSE.1987.226478]

[36] Rushby J. Noninterference, transitivity, and channel-control security policies. SRI Int'l, Computer Science Laboratory, 1992.

[37] Van Der Meyden R. What, indeed, is intransitive noninterference? In: Proc. of the Computer Security (ESORICS 2007). Berlin, Heidelberg: Springer-Verlag, 2007. 235−250. [doi: 10.1007/978-3-540-74835-9_16]

[38] Boettcher C, DeLong R, Rushby J, Sifre W. The MILS component integration approach to secure information sharing. In: Proc. of the IEEE/AIAA 27th Digital Avionics Systems Conf. (DASC 2008). IEEE, 2008. [doi: 10.1109/DASC.2008.4702758]

石勇(1982－),男,湖南桃江人,博士生,主要研究领域为可信计算,云计算安全.

韩臻(1962－),男,博士,教授,博士生导师,主要研究领域为信息安全.

郭煜(1982－),男,博士生,主要研究领域为可信计算,云计算安全.

马威(1985－),男,博士生,主要研究领域为可信计算,云计算安全.

刘吉强(1973－),男,博士,教授,博士生导师,主要研究领域为隐私保护,可信计算,云计算安全.

常亮(1980－),男,博士,教授,主要研究领域为知识表示与推理,可信计算.