# 云计算安全:架构、机制与模型评价

林 闯 苏文博 孟 坤 刘 渠 刘卫东

(清华大学计算机科学与技术系 北京 100084)

**摘 要** 随着云计算服务的广泛使用,租户对云计算的安全性提出了越来越高的要求,云计算环境的动态性、随机性、复杂性和开放性使得原有安全方案难以适用,这也阻碍了云计算的进一步发展与应用.文中在分析云计算服务模式特点以及安全挑战的基础上,针对云计算安全架构、机制以及模型评价三个方面展开研究与综述.文中指出云计算的安全架构不仅需要可信根、可信链路以及上层可信服务的安全保证,还需要实现可管、可控的动态安全管理与可度量的安全评价优化过程.文中对已有云计算安全机制和模型评价方法进行了比较分析,最后提出了基于多队列多服务器的云计算安全建模与分析思路.

**关键词** 云计算;安全架构;安全机制;安全模型;安全度量

**中图法分类号** TP393 **DOI号** 10.3724/SP.J.1016.2013.01765

## Cloud Computing Security: Architecture, Mechanism and Modeling

LIN Chuang SU Wen-Bo MENG Kun LIU Qu LIU Wei-Dong

(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

**Abstract** With the wide use of cloud computing services, tenants require higher and higher security assurance. The characteristics of cloud computing are dynamics, randomness, complexity and openness, which make the original security solutions difficult to be applied to the cloud environment, so this is a big obstacle to the development of cloud computing. We analyze the features and security challenges of cloud computing, and do a survey on security architecture, mechanism and model. We propose a security architecture which needs trusted root, trusted link and trusted high level services to be the security assurance, and it also needs controllable, monitorable security management and measurable process of security optimization. We analyze and compare the present research results of security model and mechanism in the cloud. At last, we propose a security modeling method based on the multiqueue multiserver model.

**Keywords** cloud computing; security architecture; security mechanism; security model; security measurement

## 1 引 言

云计算是继分布式计算[1]、网格计算[2]、对等计算[3]之后的一种新型计算模式,它以资源租用、应用托管、服务外包为核心,迅速成为计算机技术发展的热点.在云计算环境下,IT领域按需服务的理念得到了真正体现.云计算通过整合分布式资源,构建应

收稿日期:2013-04-22;最终修改稿收到日期:2013-07-17.本课题得到国家"九七三"重点基础研究发展规划项目基金(2010CB328105,2009CB320505)、国家自然科学基金重点项目(60932003,61020106002)和国家自然科学基金面上项目(61070182,60973144,61173008,61071065)资助.林 闯,男,1948年生,博士,教授,博士生导师,主要研究领域为计算机网络、系统性能评价、安全分析和随机Petri网.E-mail:chlin@tsinghua.edu.cn.苏文博,男,1989年生,博士研究生,主要研究方向为性能评价和云计算隐私与安全.孟 坤,男,1980年生,博士,助理研究员,主要研究方向为性能评价和随机模型.刘 渠,男,1990年生,博士研究生,主要研究方向为性能评价和并行计算.刘卫东,男,1968年生,博士,教授,博士生导师,主要研究领域为云计算架构和网络计算.

对多种服务要求的计算环境，满足用户定制化要求，并可通过网络访问其相应的服务资源. 云计算对资源共享且高效利用的特点，可以实现系统管理维护与服务使用的解耦. 如何利用云计算的相关成果促进国计民生行业的发展，已成为国家发展战略的重要组成部分.

Gartner 早在 2011 年 1 月发布的 IT 行业十大战略技术报告中就将云计算技术列为十大战略技术之首①. 据 IDC(International Data Corporation)预测，2015 年云计算服务的年收益将高达 729 亿美元；并在未来 5 年内，云计算服务仍将保持强劲的增长态势，平均年增幅将达到 27.6%，是传统 IT 行业平均增长速度 6.7%的 4 倍②. 云计算具有广泛的应用前景，Google、IBM、Microsoft、Amazon、腾讯、阿里巴巴等知名 IT 企业都在大力开发和推进云计算.

然而，随着云计算用户和服务内容的爆炸式增长，用户需求和提供商服务模式之间的矛盾日趋明显. 可见云计算要成为真正为广大用户普遍认同的信息服务基础架构还面临着诸多挑战. 根据 Gartner 的调查报告③(见图 1)，安全性、可用性和性能成为用户最关注的三个方面；另外，支付模式的经济性、系统的兼容性、现有 IT 基础设施的继承性及定制化的灵活性也有超过 75%的受访用户表示关注. 因此，不仅需要实现现有云计算架构与机制的融合，还要建立用户友好、服务管理透明、安全与性能得到实时保证的云计算标准，这些问题都将成为学术界和工业界普遍关注的研究重点.

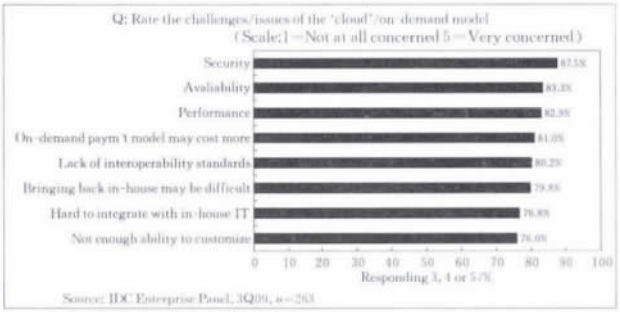


图 1 云计算发展中面临的挑战③

2007 年 SaaS 服务提供商 Salesforce.com 遭受攻击，导致大量租户的隐私数据泄露丢失④；2010 年 Google 两名员工侵入租户的 Google Voice、Gtalk 等帐户，引起隐私数据泄露；由于缺乏数据加密和分散隔离存储机制，导致了 CSDN 网站 600 多万用户的数据库信息被黑客盗取并公开. 不断发生的云计算安全事件充分说明了云计算系统已成为黑客攻击的众矢之的. 安全与隐私作为云计算租户最关心的问题，在越来越多的安全风险涌现的情况下，工业界与学术界也不断地提出了相应的安全机制和管理方法. 为了更好地认识云计算，指导租户和服务提供商恰当地选择其云计算策略，本文从云计算安全的角度来阐述和分析现有的云计算的安全架构、安全机制和安全服务模式，并在此基础上提出一种可度量的云计算安全架构，最终探讨安全量化分析模型在云计算安全中的使用方法.

本文第 2 节对云计算的服务特点、面临的安全性挑战进行分析，并给出云计算安全服务框架；第 3 节分析探讨并给出可管、可控、可度量的云计算安全架构；第 4 节根据要实现的安全目标，对目前常采用的安全机制展开分析，分类比较各种机制的应用条件和相互关系；第 5 节将结合云计算特点，在总结现有安全量化分析模型的基础上，对云计算安全度量指标、安全度量模型及计算与分析方法进行全面探讨，并给出一种基于多队列多服务器云安全模型的分析思路；第 6 节对全文进行总结，指出可行的研究方向.

## 2 云计算安全概述

该部分将首先介绍云计算及其服务模式，并在此基础上总结云计算安全所面临的挑战，探讨性地给出云计算安全服务框架.

### 2.1 云计算服务模式

网络通信、分布式计算及服务计算等技术的发展为云计算的实施提供了强有力的支撑. NIST 指出云计算是一种可以通过网络连接，便携且按需访问的可配置共享资源池的服务，计算资源将以最小的管理和交互代价快速提供给用户；同时云计算还应满足按需自助服务、广泛网络接入、高效资源共享、高弹性计算、支持度量计费等五大功能特性[4].

根据云计算所提供服务类别的不同，云计算的服务模式可以分为软件即服务(Software as a Service，SaaS)、平台即服务(Platform as a Service，PaaS)和基础设施即服务(Infrastructure as a Service，IaaS). 此外，为实现计算资源本地化，目前如 Microsoft、IBM 等公司提供的服务器集装箱租赁服

① http://www.gartner.com/newsroom/id/1454221
② http://www.idc.com/prodserv/idc_cloud.jsp
③ http://blogs.idc.com/ie/?p=730
④ http://voices.washingtonpost.com/securityfix/2007/11/salesforcecom_acknowledges_dat.html

务可以被认为是一种新的服务模式，并称之为硬件即服务(Hardware as a Service，HaaS)①.

HaaS、IaaS、PaaS、SaaS 在功能范围和侧重点上都存在差异，其中，HaaS 仅提供满足租户需求的硬件资源，包括存储空间、计算能力、网络带宽等，重点在于保证硬件资源的性能和可靠性；IaaS 需要在异构资源环境下，提供按需付费、可度量资源池功能，同时要兼顾硬件资源的充分利用和用户需求的满足；PaaS 不仅关注底层硬件资源的整合，还需要提供能够供租户进行开发、调试应用的平台环境；SaaS 不仅需实现底层资源的充分利用，还必须通过部署一个或多个应用软件环境，为用户提供可定制化的应用服务. 为叙述方便，我们把上述服务模式看成是从低到高的四个层次. 根据各种服务模式可采用的实施方式的不同，其与物理基础设施的支持关系，如图 2 所示[5]. 不难发现，较高层次的服务提供商可以独立建立服务资源，也可以借用较低层次云服务商提供的服务资源，例如 SaaS 服务可以由 Salesforce 独立提供，也可以由 SaaS 应用开发者在租用的其他 PaaS 平台上提供.

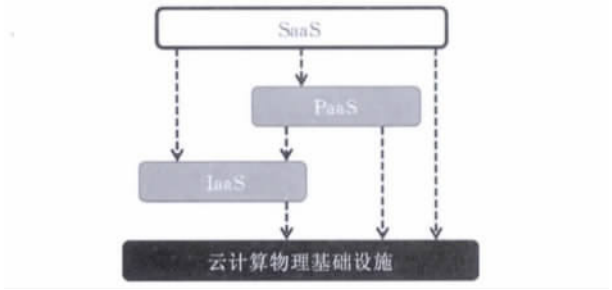


图 2　云计算服务传递模式[5]

事实上，随着服务模式层面的不断上移，服务的功能和需要满足的条件呈现递增被包含关系，如 SaaS 不仅关注底层(PaaS，IaaS)的实现，还需要考虑软件的具体功能实现和优化.

### 2.2　云计算安全挑战

云计算的主要目的在于帮助租户摆脱纷杂的硬件管理与维护，实现系统资源的深度整合. 通过统一管理模式提高资源利用率的同时，满足各类租户的个性化需求. 其实现方式决定了租户的数据信息势必会存储在公用数据中心，数据的读取完全依赖于网络传输. 因此，云计算系统不仅面临着传统信息系统(或软件系统等)的安全问题，还面临着由其运营特点所产生的一些新的安全威胁[6].

概括地，云计算在安全方面必须解决好下列问题：多租户高效、安全地资源共享；租户角色信任关系保证；个性化、多层次的安全保障机制；以及效率、经济性与安全性兼顾的多属性服务系统.

多租户高效、安全地共享资源. 资源的共享实现了服务成本下降和可扩展性提高，但同时也给安全带来了巨大挑战. 一方面，共享系统为安全风险的快速蔓延提供了条件. 另一方面，多租户共享的特征给恶意租户攻击其他租户或自私租户恶意抢占资源提供了便利. 如常见的安全威胁包括：恶意租户使用侧信道(Side Channel)方法探测运行在同一主机上其他租户的隐私数据；通过抢占大量资源致使其他租户的服务不可用等[7].

租户角色信任关系保证. 一方面，云端管理租户的定制应用，降低了租户对数据处理过程的可控性，为防止提供商利用其特权窃取租户的隐私信息，系统应具有相应机制保证租户和提供商间的可信关系. 另一方面，租户间的数据共享和传输依赖于租户设置的访问控制协议，为防止数据的恶意篡改和泄露，需要提供必要的手段保证租户间的可信关系，例如文献[8]中采用了基于声誉的用户可信管理方案. 此外，提供商对第三方软件的安全性难以实现完全认证，为防止恶意代码对系统的影响，建立相应机制保证第三方软件的可信性也是必不可少的.

个性化、多层次的安全保障机制. 根据定制的服务模式、服务内容，租户对云计算系统具有相应的安全性需求[9]. 为满足个性化和层次化的需求，云计算系统应综合考虑服务特性、复杂度、可扩展性、经济性等因素，设置具有较高灵活度且明确的安全保障机制的定制模块，为租户的选择提供便利. 一般地，各层面的保障机制应分别满足下列要求：SaaS 需要提供对企业更加透明的数据存储和安全方案；PaaS 应具有完善的访问控制机制防止平台被黑客利用；IaaS 应实现数据存储、资源利用的合理性和安全性；HaaS 则应关注硬件的性能和数据的泄露. 文献[10]从技术层面针对各层服务详细地总结了云计算面临的具体安全问题.

效率、经济性与安全性兼顾的多属性服务系统. 从租户的角度看，他们不仅关注系统的安全因素、功能实现，还要权衡服务质量、开销等因素以选择云计算服务商. 从服务提供商的角度看，其必须在安全、功能、性能等方面强化，吸引尽量多的用户使用，从而发挥数量优势降低运营成本，实现利益的最大化. 上述需要考虑的因素涉及系统的多种属性，如性能、

① http://www.roughtype.com/?p=279

安全性、公平性等，它们之间存在交叉和相互影响，兼顾多种属性的云计算系统是未来的发展方向. 因此，就云计算安全问题而言，其同样要融入多属性兼顾的服务系统中去解决. 多属性的选择及其形式化定义是指导优化设计安全机制的关键，也是建立多属性服务系统的理论基础. 该系统的理想状态应能够有效平衡系统的各种属性，避免安全瓶颈的出现.

最后，从云计算的构建和发展模式上来看，我们还必须注意以下事实：云计算服务环境的构建，不可能依靠一次性购置设备完成，势必要经历循序渐进、不断完善的发展历程. 因此，对已有资源具有良好的继承性和兼容性成为云计算发展的一个重要要求，其要求云计算安全框架、机制的设计必须要考虑资源异构性、安全差异性的影响.

### 2.3 云计算安全服务框架

针对上述云计算所面临的安全性挑战，该部分将探讨一种基于模型分析的云计算安全服务框架.

事实上，云计算可以被简单地看作一个服务系统，包括服务提供商和租户两个使用主体. 提供商运营云计算系统，通过网络发布其提供的服务内容；租户根据需求向服务提供商定制其个性化需求，并依据服务提供商给予的权限访问使用云计算系统，如图 3 所示. 目前，多个云计算服务商提供了监控和展示模块，如 Salesforce，可以实时地向用户展示系统的运行状况. 但是，其数据多为日志数据的简单展示，对于租户而言难以直接指导其行为的选择. 因此，我们认为应该在监控模块中增加模型分析模块，使整个云计算真正成为动态、可观测、可控的服务系统. 租户和提供商都能实时地得到其关心的属性指标，并分别采取相应措施保证云计算系统成为高效运行的生态系统.

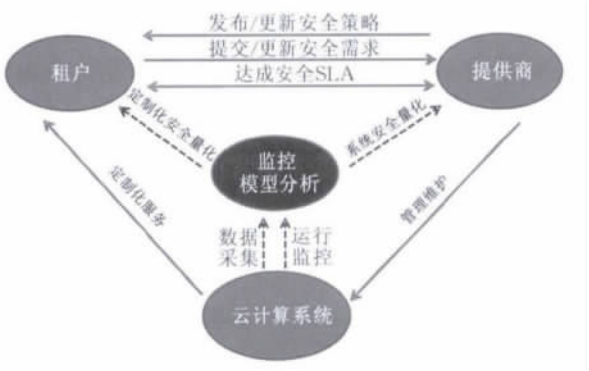


图 3 云计算安全服务框架

以云计算安全问题为例，在监控和模型分析模块中，可以针对系统的多个属性（如系统可用性、系统性能等）建立多角度的（如瞬时、稳态、区间值等）分析模型，并依据监控得到的数据向租户和服务商发布属性指标值，供他们选择相应的策略. 云计算提供商发布或者更新供租户选择的安全机制集合（或模板），并可以根据需要动态地增加和删减机制集合. 租户根据业务需求及服务提供商的可选机制集合，定制其需要的安全策略，并根据观测到的系统属性指标动态地改变其策略. 双方达成的 SLA（Service-Level Agreement）协议为双方履行安全责任的依据.

上述云计算安全服务框架中，系统的监控和模型分析是重点，处于整个安全过程的核心位置，其效果直接影响系统主体策略的选择，进而影响系统的运行效率. 提供商应针对云计算的安全挑战设计合理的安全架构、机制集合，并通过设计服务模板和激励机制引导租户合理地选择其策略.

在本文的剩余部分中，我们将在总结现有安全框架、安全机制和安全模型分析方法的基础上，进一步阐述上述云计算安全服务框架的实施方法.

## 3 云计算安全架构

在总结现有云计算安全架构的基础上，给出一种可管、可控、可度量的云计算安全架构.

### 3.1 基于可信根的安全架构

保证云计算使用主体之间的信任是提供安全云计算环境的重要条件，也是该类安全架构的基本出发点. 尽可能地避免安全威胁得逞、及时发现并处理不可信的事件是该架构的设计目的. 一方面要求包括云计算提供商在内的各主体，在时间和功能上只有有限的权限，超过权限的操作能够被发现并得到妥善处理；另一方面要求主体的使用权限在具有安全保证的前提下可以便捷地变更，针对 HaaS 服务这项功能尤其重要. 该架构的典型代表为基于 TPM 可信平台模块的云计算安全架构. 接下来从 TPM 平台模块的功能特性和可信系统的搭建两方面阐述该架构.

#### 3.1.1 TPM 可信平台模块

可信平台模块 TPM（Trusted Platform Module）是按照可信计算组织 TCG（Trusted Computing Group）制定的标准所实现的加密芯片，可以捆绑在通用硬件上，目前全球采用 TPM 芯片的终端已经达到了 19 亿[①].

TPM 安全芯片的实质是一个可独立进行密钥

① https://www.trustedcomputinggroup.org

生成、加解密的装置，内部拥有独立的处理器和存储单元. 针对云计算环境，TPM 芯片应满足以下基本要求[11]：(1) 完整性度量，保证使用该芯片的机器，从启动开始的每个操作都具有完整的验证机制，防止黑客或者病毒篡改系统信息；(2) 敏感数据加密存储与封装，将敏感数据存储在芯片中的屏蔽区，用户数据通过硬件级加密存储到外部设备，防止数据被窃取；(3) 身份认证功能，硬件级的用户身份标识，秘钥和硬件的绑定实现身份确认，防止伪装攻击的发生；(4) 内部资源授权访问，通过 TPM 的授权协议能够方便地实现用户对其资源设置访问权限，在保证安全性的情况下实现资源的便捷共享；(5) 数据加密传输，能够在与外界进行通信时，加密通信链路上的数据，防止监听、篡改或者窃取. 目前，最新发布的 TPM2. 0 标准将支持多种加密算法，提高了芯片使用的灵活性.

3.1.2　云计算 TPM 可信根架构

可信根是能够保证所有应用主体行为可信的基本安全模块，其不仅可以判断行为结果的可信性，还能够杜绝一切非授权行为的实施，被认为是构建可信系统的基础. TPM 作为目前普遍认可的可信计算模块，被广泛应用为可信计算系统的可信根[12]. 针对云计算环境，文献[13]提出了一种基于 TPM 的可信云计算平台架构 TCCP (Trusted Cloud Computing Platform)，其实施协议如图 4 所示.

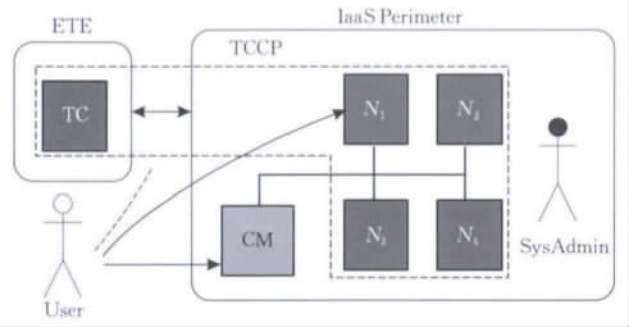


图 4　TCCP 协议示意图[13]

在 TCCP 中，云计算用户的使用空间为基于 TPM 的封闭虚拟环境，用户通过设置符合要求的秘钥等安全措施保证其运行空间的安全性；云计算管理者仅负责虚拟资源的管理和调度. 用户私密信息交由使用 TPM 的可信计算管理平台保管，实现了与云计算管理者的分离. 从而利用 TPM 实现了防止管理者非法获取用户数据、篡改软件功能的行为.

详细地，TCCP 主要包含两个模块：可信虚拟机监控模块(TVMM)和可信协同模块(TC). 每台主机需安装 TVMM 模块，并且嵌入 TPM 芯片. TVMM 可以不断验证自身的完整性，提供了屏蔽恶意管理者的封闭运行环境，而 TPM 芯片则可以通过远程验证功能，使用户确定主机上运行着可信的 TVMM 模块. TC 主要功能是管理可信节点的信息，保证可信节点嵌入了 TPM 芯片，并将认证信息通知给用户. 因此，一般由一个类似 VeriSign 的第三方可信机构进行管理.

与 TCCP 类似的云计算安全架构还包括：可将数据和操作安全进行外包的云计算安全架构[14]，其中，非可信云计算提供商负责数据的具体操作，可信云计算提供商负责事前的加密和事后的认证，整个过程中用户仅需要和可信云计算提供商进行通信. 另外还有基于两层可信传递链机制的安全架构[15]，其满足了用户控制云计算平台的需求，并通过远程验证功能确保了 IaaS 平台的可信性，减少了云计算提供商不必要的管理责任.

### 3.2　基于隔离的安全架构

租户的操作、数据等如果都被限制在相对独立的环境中，不仅可以保护用户隐私，还可以避免租户间的相互影响，是建立云计算安全环境的必要方法. 目前，基于隔离的云计算安全架构研究主要集中在软件隔离和硬件隔离两个不同的层面上. 目标在于为租户提供由底至顶的云计算隔离链路，如图 5 所示. 在该部分将针对上述两个层次的实现机理作简要介绍.

3.2.1　基于软件协议栈的隔离

针对云计算硬件资源的分布性和多自治域的特点，采用虚拟化的方法，实现网络、系统、存储等逻辑层上的隔离是该方案的主要特点.

但是，由于隔离机制涉及的环节较为分散，目前并没有达成统一的协作规范，设备和技术的差异导致无法形成高效的端到端隔离. 为确保该类方案的高效运行，不仅需要各个环节实现有效的隔离机制，还要建立隔离机制之间的协作协议.

目前，虚拟化技术的发展支持着该方案的实施，典型的代表为富士通提出的可信服务平台的统一隔离方案①. 其中，隔离的软件协议栈是指云计算各环节中隔离机制的总和，根据云计算服务过程中涉及的隔离策略，可以把其划分为服务终端、网络、系统、存储等多个环节的隔离. 通过采用统一的高级虚拟化技术，实现逻辑层的隔离，并达到与物理隔离一样

① http://www.fujitsu.com/downloads/MAG/vol46-4/paper09.pdf

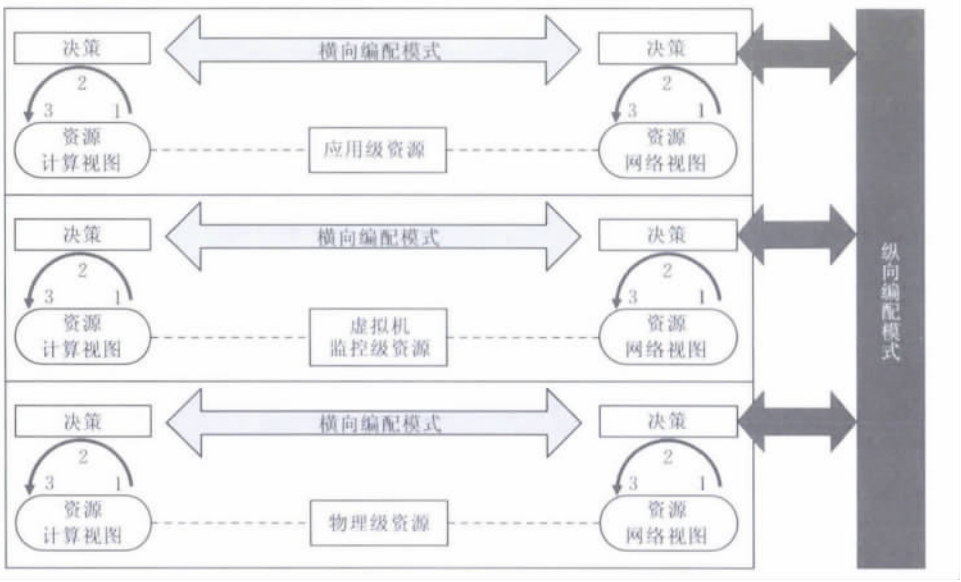


图 5 一种多层次隔离示意图[16]

安全的效果.

此外,文献[16]中将 IaaS 服务细分为三层:物理层、系统监控层以及应用和虚拟机层,并把其类比为 OSI 的网络服务层次. 通过设计不同层次隔离机制的协作方案,实现了自动化的资源隔离框架,也实现了对资源隔离的高效管理.

3.2.2 基于硬件支持的隔离

相对于通过软件实现逻辑层隔离的架构,硬件支撑的隔离方案具有更好的安全效果,并随着硬件功能的提升,也使之逐步成为了可能. 其中典型的代表为以思科为首的公司提出的安全云架构①,如图 6 所示.

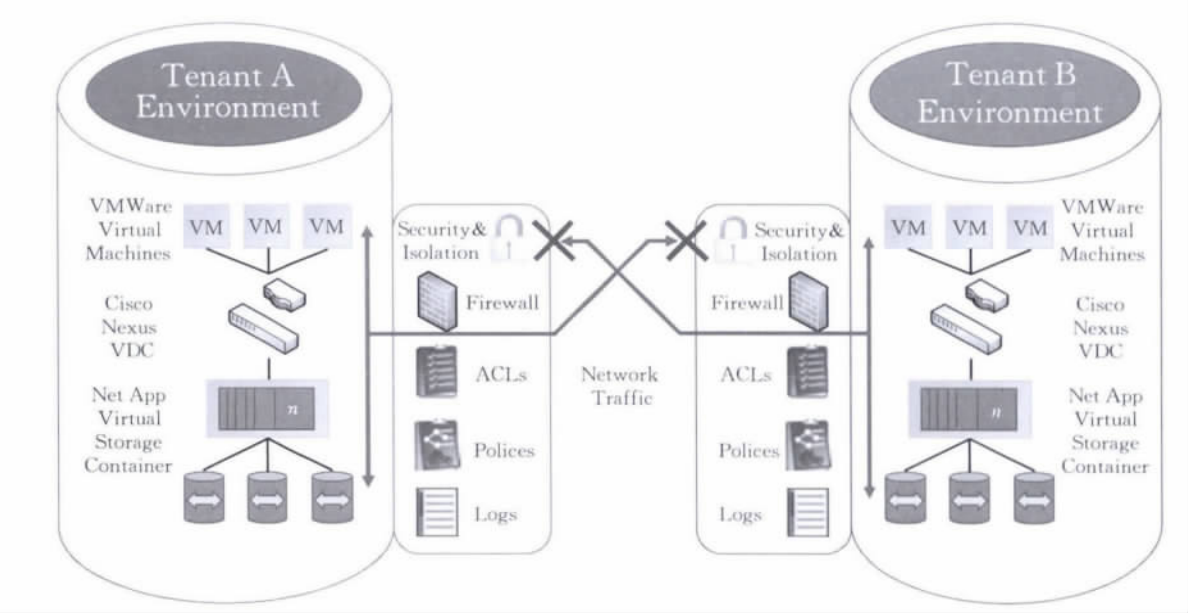


图 6 思科安全云架构

该方案由 NetApp、思科、VMware 三个公司联合提出,并针对不同的层次给出了各自的解决方案. 其中,NetApp 的 Mutistore 在独立的 NetApp 存储系统上快速划分出多个虚拟存储管理域,每个虚拟存储管理域都可以设置不同的性能及安全策略,从而实现租户在牺牲最少私有性的前提下安全地共享同一存储资源. 为了实现与隔离网络的高效对接,NetApp 的 VLAN 接口可以创立私有的网络划分,每个接口绑定一个 IP 空间,IP 空间是独立且安全的网络逻辑划分,代表一个私有的路由域.

① http://www.scalar.ca/newsevents/download/SecureCloudArchitecture-1.pdf

思科在网络隔离方面也提供了硬件支持，其交换机有能力把一台物理交换机分成至多四台虚拟交换机，每台虚拟交换机和独立交换机一样，具有独立的配置文件、必要的物理端口以及分离的链路层和网络层服务.

在应用层，VMware 的 vShield Zones 技术提供了对网络活动更强的可视性和管理能力，在虚拟服务层建立了覆盖所有物理资源的逻辑域，实现了租户之间不同粒度的信任、隐私以及机密性管理.

与传统架构相比，该架构在硬件支持的基础上，实现了对存储、网络、虚拟机、服务器各个环节的高效隔离以及高效连接，保证了多租户环境下数据的安全性以及系统的高效性，避免了以大幅度降低效率为代价的多租户隔离.

但是，由于该方案具有较强的硬件依赖性，尽管可以实现高效的链路隔离，但较高的成本使其在已有云计算环境下难以推广.

### 3.3　安全即服务的安全架构

租户业务的差异性使得他们需要的安全措施也不尽相同，单纯地设置统一的安全配置不仅会导致资源的浪费，也难以满足所有租户的要求. 目前，借鉴 SOA(Service Oriented Architecture )理念，把安全作为一种服务，支持用户定制化的安全即服务的云计算安全架构得到了广泛的关注，本节将介绍 SOA 理念在云计算安全中的含义，以及基于 SOA 的云计算安全架构.

云计算平台上运行着不同的服务，需要数据库、网络传输、工作流控制以及用户交互等多种功能的支持. 由于执行环境和执行目的的不同，必然面临不同的安全问题. 其中，数据库面临数据存放、加密、恢复以及完整性保护等问题，网络传输面临外部通信以及云环境内部通信的安全问题，工作流控制面临访问控制等问题. 因此，云计算安全架构除了需要上述讨论的可信根与隔离链路保证之外，还需要在此之上构建基于 SOA 的安全服务.

#### 3.3.1　IBM 的 SOA 通用安全架构

SOA 旨在通过将结构化的软件功能模块(也称作服务)整合在一起，以提供完整的功能或者复杂软件应用的设计方法，主要体现了服务可以被设计为具有专门功能，并且可以在不同应用之间复用的思想. SOA 希望实现服务与系统之间的松散耦合，将整体功能分为独立的功能模块，并设计模块之间规范的数据交互模式，以满足用户通过不同服务组合的方式实现定制化的服务需求[17]. GGF 指出 SOA 是一种可以用于搭建可靠分布式系统的体系结构风格，以服务的形式实现各种功能，并且强调松散的服务耦合①.

借鉴 SOA 的理念，把安全机制(或策略)看做独立的服务模块，IBM 针对云计算给出了通用的安全架构[18]，见图 7 所示. 上述架构强调云计算的各种服务模式，由于执行环境和执行目的的不同，必然面临不同的安全问题，并需要一系列具有针对性的安全机制来应对. 通过把安全机制设计为安全服务模块，可以实现不同管理域或者安全域内租户的通用性. 通过租户的选择，可以形成一个独立的云计算安全服务体系，满足租户在安全方面的个性化需求.

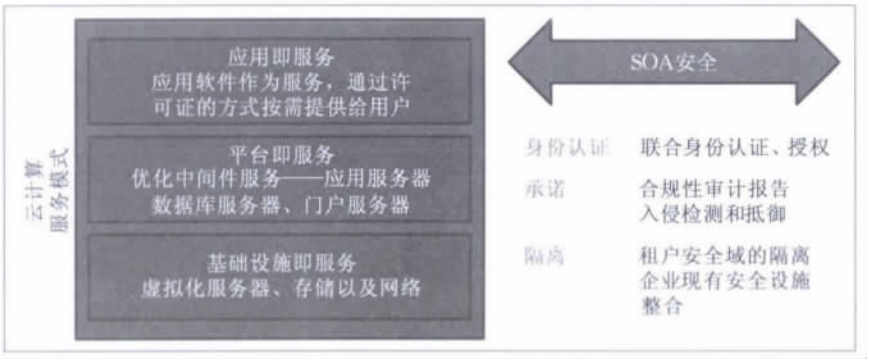


图 7　IBM SOA 通用安全架构[18]

该通用架构的主要优势在于可以轻松整合来自不同云计算提供商的安全服务. IBM 的安全架构不限制用户使用特定的安全协议或者机制，充分给予了用户灵活的选择空间，这也增加了租户对云计算提供商的信任.

#### 3.3.2　EasySaaS 的 SOA 安全方案

EasySaaS[19]是针对 SaaS 服务模式提出的一种发展框架，包括了 SaaS 系统构建、部署、更新的全过程，其强调模块化的设计理念，如图 8 所示.

① http://www.ggf.org/documents/GWD-I-E/GFD-I.044.pdf, 2005

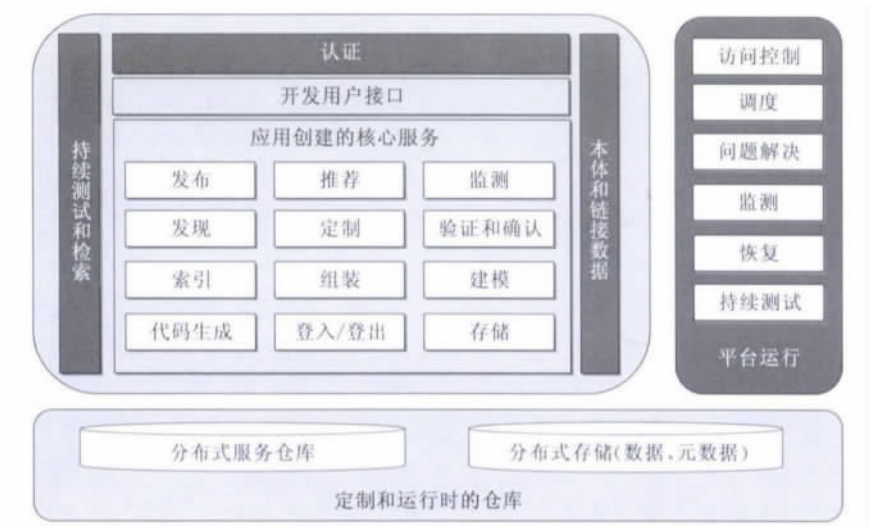


图 8 EasySaaS 架构示意图[19]

针对安全问题，EasySaaS 分为运行态和部署态两种情况分别考虑，并设计了基础安全模块（认证、测试、监控等）和定制化安全模块. 该架构完全符合 SOC(Service Oriented Computing)的要求，租户不仅可以以面向服务的方式开发、发布、发现、组装、部署和执行其个性化应用，而且还可以选择符合其安全需求的 SaaS 体系架构.

此外，该架构支持重载(heavy load)定制，租户可以选择已经存储在 SaaS 数据库里的组件，也可以开发自己个性化的组件并添加到数据库中. 租户通过链接图形界面、工作流、服务和数据组件实现组装，推荐系统可以依据租户应用和现有组件的结构、语义等信息，向租户提供合适的推荐建议. 该架构具有较高伸缩性，不仅支持简单的 SaaS 服务调用，还提供了支持代码级开发的功能.

### 3.4 可管、可控、可度量的云计算安全架构

基于可信根的云计算安全架构试图通过可信计算的成果从根本上解决云计算的安全问题. 但是，其对硬件要求的严格性有违云计算开放性和经济性的基本要求，不利于对现有资源的继承与利用. 基于隔离的云计算安全架构旨在针对所有的租户构建封闭且安全的运行环境，从而保证其定制服务的安全性. 但是，其势必导致资源的不充分利用，增加租户间协作的难度，引起管理成本的提高. SOA 架构充分考虑到了租户的个性化需求，提出以租户服务要求为导向的云计算安全架构，但是缺乏让租户和提供商及时且明晰地获得各自安全需求的方法. 基于此，通过吸取上述架构优点，我们给出了一种基于云计算安全模型评价的可控、可管、可度量的安全架构，上述云计算安全架构从不同角度阐述了保证云计算安全环境的机理，但如何让云计算主体感知其安全性需求，并及时选取相应的机制仍面临巨大挑战. 该部分将探讨这种可管、可控、可度量的云计算安全架构，如图 9 所示，并从其架构组成和实施流程两方面加以说明.

#### 3.4.1 安全架构组成分析

该架构的设计借鉴了 SOA 思想，主要包括三个组成部分：云计算安全服务框架、云计算安全技术框架和云计算安全度量框架. 安全服务框架主要用于实现租户的定制化需求，是租户安全目标的集合；安全技术框架负责管理各种云计算安全机制，也是架构安全方法的集合；安全度量框架提供系统的安全状态分析，为租户和提供商选取安全策略提供数据支撑.

针对安全服务框架，应能够尽可能全面地反应云计算服务的安全需求，如 SaaS、PaaS、IaaS 等 3 种服务模式中的安全需求. 其次，针对各种服务模式的安全需求，应能够随着租户要求的变化自适应地提供多种安全方案. 通过进一步分析可知，云计算为了支持上层服务，还需要从下至上保证五个层次的安全：物理设备的安全；网络、服务器以及终端的安全；数据以及信息的安全；租户身份授权与认证的安全.

安全技术框架需要尽量全面地包含已有的安全机制供租户选用. 该安全技术框架的设计参考了可信系统[20]的建立，首先要拥有基于 TPM 功能的可信根保证，然后在此基础上建立一条基于隔离的可信链路，最后再将可信链路传递到系统的各个安全模块. 其中，安全模块的组织与实现，可以借鉴 SOA 的理念，应具有融合各种机制的能力，不仅能够满足云计算系统的安全要求，还能兼顾租户的个性化

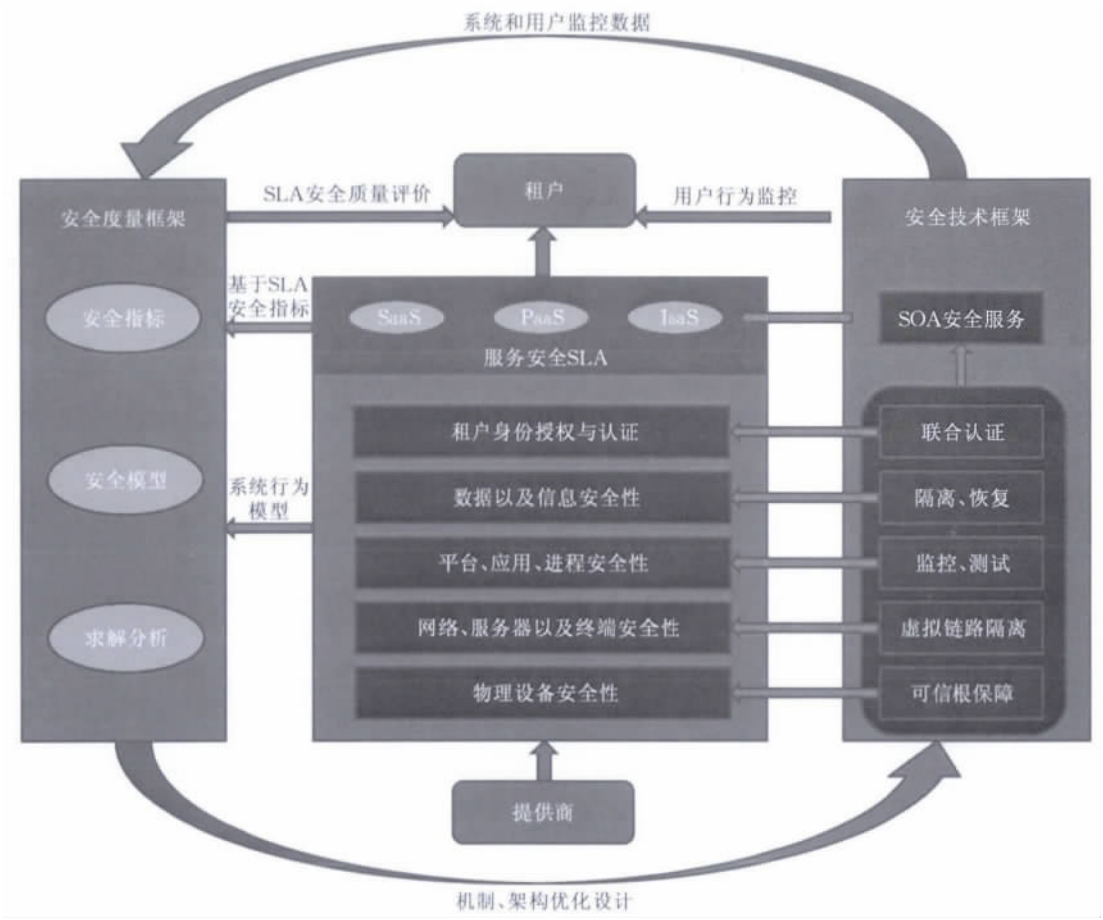


图 9　可管、可控、可度量的云计算安全架构

需求.

安全度量框架主要提供基于监控数据的安全量化分析结果，供租户和提供商选择其安全策略. 这里主要强调安全指标的可度量，并通过相应的指标形式化定义及模型求解分析予以实现.

3.4.2　安全架构流程分析

上述基本模块是该架构的安全基础，还需要架构的运行流程将各个模块形成有机的整体. 由于静态地选取上述安全技术难以保证系统服务过程中的完全安全. 因此，该架构需要提供基于度量指标的管理与控制功能，通过不断完善与维护的方法实现其安全需求. 具体流程如下，首先参考 SLA 确定系统的度量指标体系，然后利用模型分析技术，建立系统行为和用户行为的安全模型，通过模型的分析和求解，获得系统需要完善与维护的安全问题以及进一步的安全优化方案. 这里系统的监控机制不仅可以获得系统当前的状态，还可以获得系统和用户的统计数据，这些数据将是安全度量的基础.

总之，模型分析可以评价系统在攻击环境下的安全程度，帮助确定系统的安全漏洞，指导安全机制的优化设计，并且可以用较小的代价评价新安全机制的效果. 这里以安全服务框架为核心，安全度量框架和安全技术框架可以形成一个良性的闭环循环，实现系统安全性的不断优化.

## 4　云计算安全机制

根据云计算应用的全生命周期，我们把云计算安全机制分为上线前、使用中、故障修复三个阶段，分别从安全测试机制、认证与访问控制机制、安全隔离机制、安全监控机制、安全恢复机制 5 个方面展开分析.

### 4.1　云计算安全测试机制

测试与验证是及时发现安全隐患与缺陷的有效手段之一，常应用于服务上线前或运行中. 传统软件的安全测试已被证明是极具挑战性的问题，云计算环境的复杂性更加剧了测试的难度. 测试也越来越受到工业界与学术界的关注. 这里将给出目前适用于云计算安全测试的方法.

4.1.1　基于 Web 的测试机制

云计算作为基于网络的服务系统，Web 应用是

其展现服务的重要手段之一，通过借鉴传统软件的Web测试方法，目前已形成了多种针对云计算应用的测试方法.上线前，基于模型的Web测试方法往往采用工作流的方式构建测试与分析模型，实现代码的安全性测试，典型的代表包括基于Petri网的形式建模和分析技术、模型检查技术、基于模型的Web测试技术[21]等；基于测试用例的方法往往采用渗透测试来验证代码的安全性，典型代表包括多层次可信的软件错误定位方法[22]、面向可信性的测试策略和集成方法[23].运行中，主要通过实时的数据收集采用统计或基于模型的方法进行测试，其中方法[24]是基于测试和验证数据进行可信性评估和预测的典型代表.

4.1.2 增量测试机制

增量测试方法常被应用于复杂软件系统的测试与分析，通过设计合理的测试流程降低测试的复杂度.云计算的如下特征使得该测试方法在云计算中得到了广泛应用：(1)云计算系统的复杂性导致对其进行全面的测试是一件费时费力的工作，且难免出现遗漏；(2)云计算租户规模巨大，不当的测试操作会影响租户的使用体验，也可能引发严重的安全事故.

常采用的增量测试方法包括基于功能模块的增量测试和基于测试范围的增量测试.对于前者，在确定需要测试的功能模块后，在测试期间可以保证云计算服务是不间断的，并以在线的方式进行不断的验证和测试，记录用户访问和数据更新情况，并作为测试用例实时地进行安全分析和测试，通过分析得到的潜在漏洞采用放大测试和修补测试的方法，进一步确认其存在性和修补措施[25].对于后者，主要通过从有限范围内测试逐步扩展到全局测试的思路实现，该机制的典型代表为增量发布方法，即新版本首先在小范围的用户中使用，只有当小范围用户使用成功后才逐步扩大新版本的使用范围[26].

4.1.3 自动化测试机制

提高测试效率是该机制的主要出发点，通过开展深入的理论研究，设计更为智能的测试方法是提高测试效率的主要途径.针对云计算服务来源多样、定制灵活等特点，文献[27]探讨了微软与谷歌中开发人员和测试人员的比例，并根据业务的不同分析了人员分工比例，指出由于业务类型和测试重点的不同，各种业务可以采用不同程度的自动化或工具辅助的测试.文献[28]提出了将自动化自测试方法向云计算环境中迁移的思路，对云计算环境中的代码测试方法进行了分析，为实现云计算的在线测试提供了理论基础，文献[29]提出了模型驱动的云计算安全测试方法，通过建立基于UML概念的系统模型，可以实现自动风险分析以及错误用例消减.另外，针对云计算测试集合规模巨大的问题，设计高效的测试代数是提高测试效率的重要方法，亚利桑那州立大学正在研究一种名为TA的测试代数，可以识别SaaS组合测试中的错误实例，减少了测试空间，并且该测试代数可以通过并行化提高测试速度.

4.2 云计算认证与授权机制

有效的认证与授权机制是避免服务劫持、防止服务滥用等安全威胁的基本手段之一，也是云计算开放环境中最为重要的安全防护手段之一.该类机制从使用主体的角度给出一种安全保障方法，这里从服务和租户两个角度说明了该机制的主要实施方法.

4.2.1 服务为中心的认证授权

把使用服务的权限分为不同的角色，通过设置相应的认证完成对请求身份的验证和授权是该机制的基本思想.该机制在所有的云计算服务中都会被应用，其基本流程如下：云计算提供商接收到服务请求时，询问租户是否有有效的身份信息，如果有则提供完整的认证与授权管理；否则，需要建立身份信息和认证授权信息.租户通过Internet访问云计算服务，其认证过程就像使用一套独立的软件系统.整个过程中，云计算提供商可以独立完成认证，也可以委托或利用第三方机构的系统完成认证.具体认证授权过程中，每个租户通常由多个用户组成，尤其在SaaS应用中，认证授权需要应用到用户级别.租户管理员根据用户业务功能确定其角色.用户在访问租户选择的功能模块时，需要根据角色获得对应的权限.针对上述特性，文献[30]提出了一种基于RBAC(Role-Based Access Controls)[31]模型的多租户访问控制解决方案，可以解决多租户之间角色冲突、租户访问控制异构等问题.

4.2.2 租户为中心的认证授权

云计算环境中，租户可能定制来自不同管理域的服务，仅采用基于服务的认证和授权，势必导致其认证过程的繁琐，影响租户的使用体验.基于租户的认证旨在简化该过程，通过联合认证的方法在保证安全性的同时提高租户的使用体验.具有跨级跨域等特点的云计算服务，需要把用户的身份信息交由可信的第三方维护管理，所有签约服务都采用租户唯一绑定的身份和授权信息实现服务的提供[32]，最

大程度地消除租户拥有多个账号和密码可能造成的安全隐患. 目前，OpenId[①] 和 SAML[②] 等身份认证协议已经可以实现上述功能，并且得到了广泛应用. Oauth[③] 作为 OpenId 的补充实现了更为灵活的跨域访问控制，用户只需要提供一个令牌而不是用户名和密码，就可以授权服务在特定时间内访问其他服务的部分信息，进一步简化了认证流程.

### 4.3　云计算安全隔离机制

隔离一方面保证租户的信息运行于封闭且安全的范围内，方便提供商的管理；另一方面避免了租户间的相互影响，减少了租户误操作或受到恶意攻击对整个系统带来的安全风险. 这里将重点阐述网络和存储的隔离机制.

#### 4.3.1　网络隔离机制

网络隔离可以提供基本的安全保障、更高的带宽分配以及针对性的计费规则和网络的层次化支持. 网络隔离可以从物理和逻辑两个层面得以实现，其中，物理隔离需要网络设备的支持，基于 Open vSwitch 和 CiscoNexus 1000v 的物理交换机的虚拟化功能，除了实现通信隔离之外，还实现了访问控制管理等安全保障. 对于逻辑层的隔离机制，针对不同的网络协议层，可以给出了多种实现方式，例如文献[33]从网络应用层提出了一种实现隔离的方案 VIOLIN.

更多针对云计算环境的网络隔离机制，可以参考文献[34]的网络虚拟化技术综述. 但是，已有的网络隔离技术中，较少考虑来自外部或者租户之间的网络攻击，故具有更高安全性的隔离机制仍需进一步地研究.

#### 4.3.2　存储隔离机制

云计算的多租户特性对存储的隔离带来了新的挑战，云计算环境的底层模块在设计时并没有为多租户、可重配置的平台和软件提供充分的安全保障. 首先，在存储设备上，内存和硬盘等资源的再分配过程中，云服务提供商一般不会完全擦除其中的内容，那么后一个租户就有可能恢复出前一个用户的内容，造成租户隐私泄露[④]. 在存储逻辑上，ORACLE 在文献[⑤]中提出数据库隔离需要考虑安全、操作、容错以及资源等多方面因素. 因此在多租户模式下，需要重新考虑各种数据库存储模式(SQL 型、NoSQL 型、XML-based 型等)的效果. 更加具体地讨论，是每个租户应该拥有一个数据库，还是所有租户共用一个数据库，甚至是每个租户仅拥有数据库表中的若干行？虽然这些方式都可以支持多租户特性，但是其安全性以及效率各不相同. 理论上每个租户都有独立的数据库和定制化存储模式是最好的安全方式，但是云计算还需要考虑资源的利用率，NIST 在文献[35]中从 SaaS 服务的角度定性地分析了两种存储模式隔离与效率之间的平衡关系. 另外，微软定量地研究了数据从完全隔离到共享这一连续过程中的安全和效率情况，并在考虑了一些技术和商业因素的基础上，提出了可扩展的安全数据存储模式[⑥].

### 4.4　云计算安全监控机制

监控是租户及时知晓服务状态以及提供商了解系统运行状态的必要手段，可以为系统安全运行提供数据支撑. 常见的监控机制包括软件内部监控和虚拟化环境监控两种，下面分别对其进行介绍.

#### 4.4.1　软件内部监控机制

云计算的动态环境下，软件的安全问题不应该依赖事后的被动响应，需要从事后维护向事前设计、主动监控转移，形成动态的安全控制方法. 与传统软件运行环境不同，云计算分布式、去中心化的等特性对软件监测技术带来了挑战，研究者致力于开发切合这些特性的高效软件监控技术. 佐治亚理工大学提出软件断层(Software Tomography)技术，将复杂的分布式软件监控任务分解，划分给同一监控目标的多个实例，从而有效减少单一监控目标由于监控所带来的性能损失[36]；美国俄亥俄州立大学 Issos 系统提供了对并行(多处理器)和分布式(集群)系统的运行时监控支持，用户能够为监控操作定义时间约束，并能够更改运行时监控的属性值[37]. 云计算环境下典型的研究实践还包括 Google 的 Dapper[38] 和 Berkeley 的 Pinpoint[39]. 它们在软件和通信协议中注入探针，通过跟踪运行时软件调用路径的方法，进行系统性能瓶颈的分析，这样的方法同样适用于安全问题的监控.

#### 4.4.2　虚拟化环境监控机制

云计算 IaaS 服务中，租户的使用权限更高，提供商难以对租户的行为进行管理，因此需要针对虚拟化的监控与分析机制. 基于虚拟化技术的监控分析方法包括静态分析监控方法[40-42]和动态分析监控方法. 前者可以将安全监控工具置于独立的受保护空间中，从而保证监控的良好运行，但是静态检测分析方法不能对操作系统的行为，即事件操作进行监

① http://openid.net/
② http://www.webopedia.com/TERM/S/SAML.html
③ http://oauth.net/
④ http://www.r3datarecovery.com/Data_Recovery_Solutions/
⑤ http://www.oracle.com/technetwork/database/database-cloud/isolation-in-pvt-db-clouds-1652083.pdf
⑥ http://msdn.microsoft.com/en-us/library/aa479086.aspx

控.另外,动态分析监控方法又分为修改操作系统内核的和无需修改系统内核的两类方法[43-45].前者对事件行为的监控可通过在操作系统中植入钩子实现,当触发钩子时,钩子中断系统并进行相关操作.但是,这种分析监控技术最大的问题是需要修改系统内核,带来了许多不便.无需修改系统内核的动态分析监控方法中,一些方法通过跟踪系统进程间的信息流,进行入侵检测[46]和病毒清除[47],另一些方法[48-49]利用跟踪可疑来源数据(如网络数据)导致的控制流变化进行检测.

### 4.5 云计算安全恢复机制

恢复(或修复)机制是保证服务可靠性和可用性的重要手段,是典型的事后反应机制.根据其涉及的范围可以分为整体恢复机制和局部恢复机制,该部分将对其进行简要分析.

#### 4.5.1 整体恢复机制

系统恢复是恶意软件防御中的一个重要研究方向.许多恢复方法关注的重点在于恢复持久性数据和删除恶意软件.文献[50]中设计的 Taser 系统可以在发生攻击或者本地故障之后选择性地恢复合法的文件,通过自动解析规则克服了无法准确确定污染操作及其影响的问题.文献[51]提出了文件块级的语义分析方法和语义跟踪重放方法,不仅可以帮助用户更好地理解文件系统行为,也可以实现文件系统的恢复.然而在这些方法中,所有服务的状态在恢复后都会丢失.因此,现有的系统整体恢复技术会影响所有的进程,失去未感染进程实体的状态信息,导致未感染的服务进程失效,并失去用于保持业务连续性的信息,可见整体恢复技术难以适用于基于 MTA(Multi-Tenant Architacture)多租户特性的云计算服务.

#### 4.5.2 局部恢复机制

目前,针对局部恢复方法的研究,主要关注进程级实体的恢复,虽然可以克服对其他进程的影响,但是这类方法往往忽略了恶意软件执行后的次级攻击,导致受到攻击的进程恢复后,其他进程仍面临受损的风险.此外,局部恢复方法专注于进程级实体,不关注系统内核被破坏的情况,这使得系统可能面临更大的风险.对于局部恢复技术,文献[52]提出了一项创新的安全技术 RX,可以从确定性和非确定性的 bug 中快速恢复程序,并且可以提供 bug 的诊断信息;文献[53]提出的 Flashback 是一项轻量级细粒度的回滚和重放技术,通过记录进程内存状态以及进程与系统的交互关系实现回滚和重放.

因此,云计算环境的安全恢复机制需要针对进程级、软件级、平台级和基础架构级不同的服务层次,应用相应的恢复技术.通过确定不同级别程序和租户行为的依赖关系,实现更加高效的系统恢复.

上文分类总结了现有安全机制,介绍了一些具体的研究成果.上述安全机制以及思想可以解决多方面的安全问题,而且安全问题的解决也需要多种机制的配合.因此,针对各类安全机制适用的问题范围,表 1 进行了总结,旨在帮助租户更加有针对性地进行安全机制的选择.

表 1 安全机制适用范围示意

| 机制 | 分类 | 代表文献 | 适用范围 | | | | |
|---|---|---|---|---|---|---|---|
| | | | 身份认证 | 数据信息 | 平台应用 | 网络服务器 | 物理设备 |
| 测试 | 基于 Web 测试 | [21][24] | √ | √ | √ | | |
| | 增量测试 | [25] | | √ | √ | | |
| | 自动化测试 | [28][29] | √ | √ | √ | √ | √ |
| 认证 | 应用中心认证 | [30] | √ | √ | √ | | |
| | 用户中心认证 | ①② | √ | | | | |
| 隔离 | 网络隔离 | [33] | | √ | √ | √ | |
| | 存储隔离 | [35] | | √ | √ | | |
| 监控 | 软件监控 | [36][37] | √ | √ | √ | | |
| | 虚拟化监控 | [39][42] | | | √ | √ | √ |
| 恢复 | 整体恢复 | [50][51] | | √ | √ | √ | √ |
| | 局部恢复 | [52][53] | | | √ | √ | |

注:① http://openid.net/; ② http://oauth.net/

## 5 云计算安全的模型评价

云计算安全模型是实现可管、可控、可度量云计算安全的核心技术,该部分从云计算安全指标体系、指标的形式化描述、安全模型与分析方法等方面展开论述,并提出了一种以多队列多服务器模型为基础的云计算安全建模方法.

### 5.1 可度量的指标体系

云计算安全性的研究不应单独从传统安全角度

出发，作为一个服务系统，需要考虑全方面的服务质量. 事实表明，将安全包含在可信性里进行研究，不仅有利于明确安全与系统其他属性之间的关系，还可以制定平衡性更好的安全策略. 因此，下面将首先从可信角度出发，介绍系统的可度量指标体系，再进一步讨论云计算环境下安全性的新特点以及与其他可信属性的关系.

系统的可信性一直是工业界和学术界关注的热点，文献[54]把可信系统定义为：在出现人为和系统错误、恶意攻击以及设计和实现缺陷的情况下，系统仍可以按预期完成任务，可见安全性是系统可信性的一方面体现. 文献[55]总结了可信系统应关注的问题，并讨论了相关可信性指标，从系统的角度看，可信指标可以归纳为：可用性(Availability)，强调正常提供服务的能力；可靠性(Reliability)，强调提供正确服务的连续性；可生存性(Survivability)，强调抵御非正常操作的能力，可以用在遭受攻击、故障或意外事故时，仍能提供关键服务的能力来描述；可维护性(Maintainability)，强调更新和维护的难易程度和影响范围；可扩展性(Scalability)，强调应对负载变化自我调节和适应的能力；安全性(Security)，强调抵御恶意攻击的能力，常从机密性(Confidentiality)、完整性(Integrity)和可用性(Availability)的变化来衡量安全性的高低，其中机密性指系统遭受攻击的情况下信息不被未授权的用户获知的能力，完整性指遭受攻击情况下系统不被篡改和替换的能力. 上述属性关系示意图如图 10 所示.

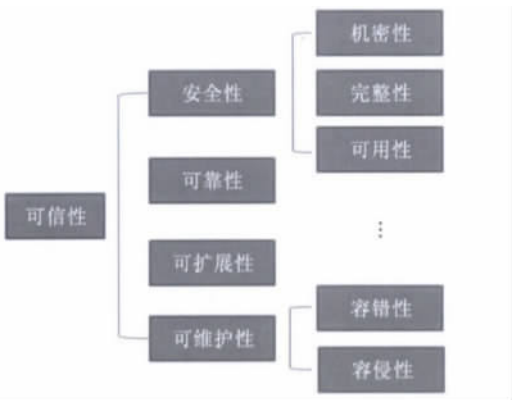


图 10　可信指标体系关系示意图

系统的安全性是可信性的重要组成部分，与可信其他属性之间存在着对立与统一的关系，在云计算环境中这种现象更加突出. 由于传统系统是相对封闭、独立的，系统的拥有者即是系统的使用者，系统提供者与系统使用者所关注的安全性指标基本一致. 然而云计算维护和使用的相互分离，使得安全性与其他可信属性的矛盾更加突出，例如，可扩展性的提高可能对安全造成严重的威胁，隔离性虽然强化了安全却也降低了系统的效率. 因此从服务提供商的角度看，其不仅要关注传统系统所需的安全指标，还需从租户的角度来扩充更为丰富的安全指标，既要考虑全体租户的基本安全保证，还应兼顾任意个体的不同安全需要，公平性(Fairness)、隔离性(Isolation)[56]等也成为必须考虑的指标. 文献[57]分析了影响租户选择云计算服务的具体因素，但是目前针对云计算系统的安全性指标体系的研究仍然不够完善.

### 5.2　安全指标形式化描述方法

目前，云计算系统提供了实时监控功能，系统维护人员可以实时获得系统运行状态，合理的安全指标形式化描述需要支持平均值、即时值的度量需求，也要涵盖不同角度、不同粒度以及多维度分析需要，可以方便系统维护人员优化系统安全策略.

云计算环境中可度量指标的形式化描述，需要根据系统具体的运行状态求得. 文献[58]提出了从用户角度出发的指标形式化描述方法，文献[59]进一步从服务计算的角度提出了系统可信性的分析方法，其具体过程如下所述：根据不同系统的运行特点，抽象出系统经历的主要运行状态，通过分析系统中的相关事件，可以获得各个状态之间的转移关系，最后获得如图 11 所示的系统状态转移模型. 模型中系统的状态可以分为 Operational 和 Failed 两大类，其中 Operational 状态又可以细化为 Recovering 和 Ready 等状态. 系统处于 Accessible 状态时，虽然完整性完好，但是可能遭到 DOS 等的安全攻击并转移到 Inaccessible 状态中. 另外需要攻击移除等安全恢复机制使系统重新转移到 Accessible 状态，因此可以通过 Attack Removal 或者 DOS 攻击等事件将两个系统状态连接起来. 系统状态转移图描述了系统安全的相关状态以及事件，不同的安全指标对应了系统不同的状态集合，据此可以获得该系统的安全指标形式化描述.

以可用性为例，可用性的定义为一个特定时间内系统能够安全运行的时间比率. 可用性是系统安全状态变化过程的一种量化描述，明确描述了系统在自身缺陷或者外部攻击情况下仍然能安全运行的可能性. 从定义出发根据系统状态转移图，可以进一步确定其数学描述，可用性一般分为稳态可用性和瞬时可用性. 假设 $S_A$ 为系统安全状态的集合，则系统瞬态可用性，即系统瞬时处于安全状态的概率为

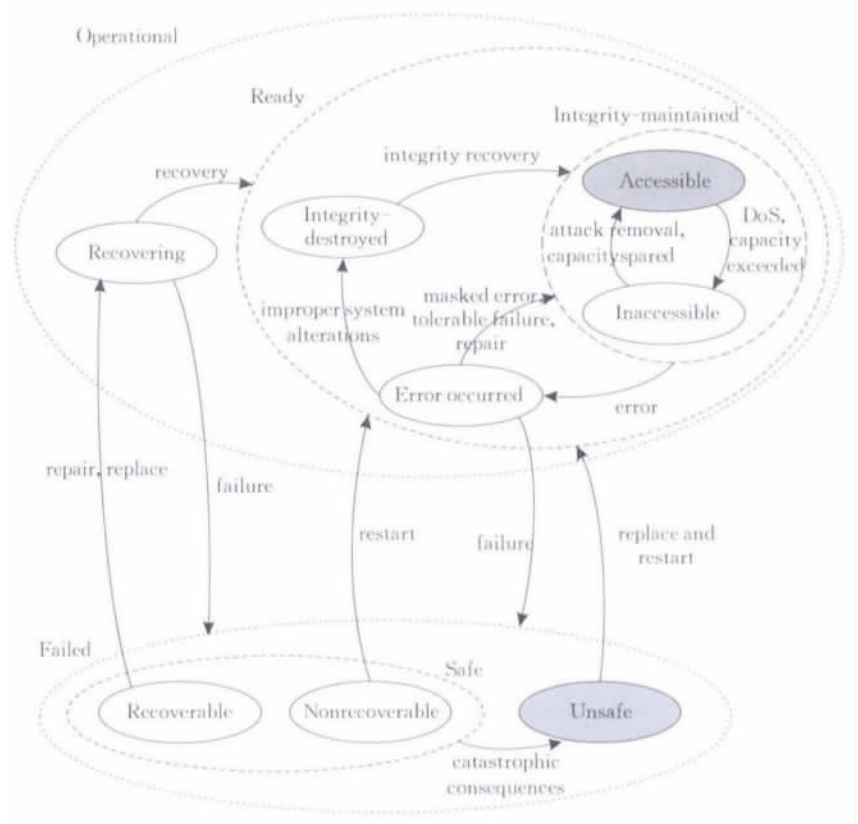


图 11 系统运行状态转移图

$$A_I(t)=P\{X(t)\in S_A\}.$$

对于系统来说，系统的稳态可用性更加重要，即稳定状态下系统处于安全状态的概率：

$$A_S=\lim_{t\to\infty}A_I(t)=\lim_{t\to\infty}\frac{\int_0^t A_I(u)\mathrm{d}u}{t}.$$

另外，还可以通过求解状态转移图中状态的稳态概率向量获得，其中 $\pi_i$ 为系统处于 $i$ 状态的稳定状态概率，则稳态可用性可以表示为

$$A_S=\sum_{i\in S_A}\pi_i.$$

针对其他安全相关指标的形式化描述，文献[60]中给出了具体的公式，在表 2 中将其进行了总结.

表 2 系统角度指标形式化描述

| 概念 | 定义 | 指标形式化公式 | 含义及变量说明 |
|---|---|---|---|
| 可靠性 | 网络系统在一个特定时间内能够持续执行特定功能的概率 | $X(0)\in S_R$，$\tau=\inf\{t:X(t)\notin S_R\}$<br>则 $R(t)=P\{\tau>t\}$ 或者<br>平均失效时间 $MTTF=E[\tau]$ | 假定 $S_R$ 为网络可执行的特定功能集合，$X(t)$ 表示在时刻 $t$ 网络系统的状态可靠性，$R(t)$ 表示系统在 $[0,t]$ 时间内持续执行功能 $S_R$ 的概率 |
| 可用性 | 特定时间内系统能够安全运行的时间比率 | $A_I(t)=P\{X(t)\in S_A\}$<br>$A_S=\lim_{t\to\infty}A_I(t)=\lim_{t\to\infty}\frac{\int_0^t A_I(u)\mathrm{d}u}{t}$<br>$A_S=\sum_{i\in S_A}\pi_i$ | $S_A$ 为系统安全状态的集合，$A_I(t)$ 为瞬时可用性，$A_S$ 为稳态可用性，$\pi_i$ 为系统处于 $i$ 状态的稳定状态概率 |
| 保险性 | 周期时间内系统不对环境和用户造成灾难性后果的概率 | $S=\sum_{i\in S_W}\pi_i$ | $S_W$ 为系统正常运行状态集合 |
| 机密性 | 机密性指系统信息等不被未授权的用户获知 | $C=\sum_{i\in S_C}\pi_i$ | $S_C$ 为满足机密性的状态集合 |
| 完整性 | 完整性理解为不出现错误的系统变化 | $I=\sum_{i\in S_I}\pi_i$ | $S_I$ 为满足完整性的状态集合 |

### 5.3　安全模型与分析

云计算系统规模增长迅速，由于分析效率、分析成本等因素的影响，通过部署和测试进行系统安全性分析的方法越来越不能满足目前云计算发展的需要. 与此同时，基于数学模型的分析方法越来越得到人们的重视，成为了分析系统特性的重要方法，也出现了众多有效的分析模型. 经过近 40 年的发展，根据分析呈现结果的不同，分析模型被分为定性分析模型和定量分析模型，定性分析模型已被成功应用于系统架构分析、系统功能分析等方向，已形成了较多的成熟模型；而定量分析模型发展则相对滞后，但由于其在策略选择、SLA 制订等方面的基础性地位，该方面的研究工作成为模型分析领域的热点.

安全分析模型主要可以分为基于组合的分析模型和基于状态的分析模型. 基于组合的分析模型更适合于系统的静态分析，无法刻画系统的随机变化，适合于系统前期设计使用. 基于状态的分析模型可以引入随机过程，模型更加接近系统实际运行过程，其度量结果可以反映系统实际运行情况. 因此，本文将主要讨论云计算环境下基于状态的安全分析模型.

#### 5.3.1　基于组合的安全模型

常见的组合模型分析方法包括：可靠性框图法、故障树分析法、模型检测分析法、攻击树分析法和攻击图分析法等，其中模型检测技术可以自动生成攻击树和攻击图. 简单的攻击树模型如图 12 所示[60]，以 Unix 系统入侵为例，入侵者首先获得远程或者局域的访问权，利用密码文件对系统密码进行破解，每个非叶子节点都是入侵子目标，可以根据成功概率和难度进行赋值，最后可以递推入侵成功的概率. 针对不同安全问题以及安全环境，组合安全模型有着不同的研究成果，文献[61-62]直观地描述了系统各组件间的逻辑关系，在已知系统组件可靠性随线性时间变化情况的前提下，分析了系统可靠性的变化；文献[63-64]由各种逻辑门组成的树状的结构描述了基本组件与系统之间故障的逻辑关系，在假设故障存在概率的前提下，分析了失效的概率；而文献[65-66]则是在已知攻击集合的前提下，综合考虑功能模块间的关系构建了对应的攻击树和攻击图模型，不仅分析了系统被成功攻击的可能性，还得到了针对性的保障措施. 文献[67]用模型检测方法成功进行了软件安全性和可靠性的缺陷检测，文献[68]给出了软件模型检测最新进展的综述.


图 12　Unix 系统的简单入侵攻击树模型[60]

#### 5.3.2　基于状态的安全模型

基于状态的分析模型包括马尔可夫链、马尔可夫回报模型、马尔可夫更新过程、补充变量分析法、随机 Petri 网[69-70]等. 马尔可夫过程是基于状态分析模型的数学理论基础，在系统安全性分析方面发挥了巨大作用，图 13 是一个基于马尔可夫过程的 DOS 攻击模型，攻击过程可以描述为四个状态，在给定状态之间转移速率的条件下，利用马尔可夫过程求得稳态概率分布，并由此进行系统的安全评价[60]. 另外，Petri 网模型由于其具有精确描述系统并行、异步、分布等特性的优势，在软件可达性、死锁、功能验证等方面有着广泛的应用. 另外，引入随机延迟时间后的 Petri 网，称随机 Petri 网，不仅继承了 Petri 网在模型描述方面的优势，还具备了随机量化分析能力，在与随机过程对应转化关系的支撑下，成为量化分析包括可用性、可靠性及安全性的有力工具.

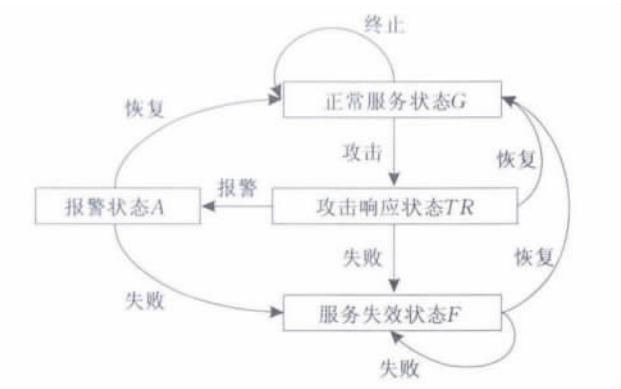


图 13　基于马尔可夫过程的 DOS 攻击模型[60]

为解决复杂化的系统安全问题，基于状态的安全模型在建模与求解上都取得了新的进展. 文献[71]中构建了一个多队列模型来整体地描述三种攻击，并研究了系统安全的性能度量，有系统可用性、平均队列长度与信息泄露可能性. 文献[72]涉及到量化不同容侵系统安全性的问题，安全入侵事件以及入侵响应被模型为随机过程，通过分析和量化系统的安全属性，可以求得系统的平均安全失效时间(MTTSF)，以及不同安全属性对安全事件的影响程度.

### 5.4 基于多队列多服务器的云安全模型

云计算的复杂性导致其安全模型难以描述，这里介绍一种基于多队列多服务器的云计算安全模型方法，利用该方法可以针对具体的云计算安全问题建立相应的安全模型.

#### 5.4.1 云计算多队列多服务器模型

云计算有着规模巨大的服务器集群，可以提供多种类型的服务. 这里介绍一种针对此种服务模式的建模方法，基于随机 Petri 网的多队列多服务器模型[73]，如图 14 所示. 在模型结构上，多队列多服务器模型中多个队列对应多种业务流，多个服务器对应异构的资源. 在模型语义上，提出变迁函数的概念，通过将变迁函数应用于 SHLPN 中的实施条件函数和实施速率函数来描述服务的不同情况. 基于 SHLPN 的多队列多服务器模型是描述分布或并行的计算机资源管理系统的一个有效的统一框架.

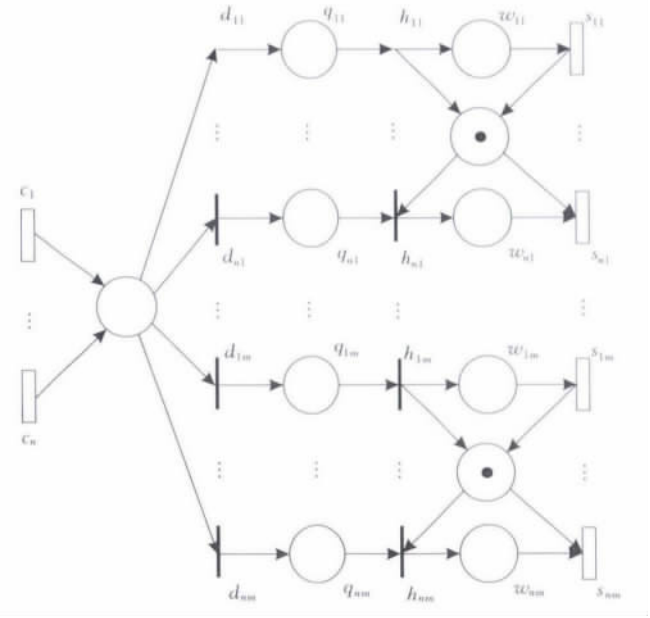

图 14 多队列多服务器模型[73]

#### 5.4.2 多队列多服务器的安全模型

借助多队列多服务器模型的描述和分析能力，可以对云计算系统中的安全问题进行更加好地建模与分析. 研究思路如图 15 所示，将多队列多服务器作为云计算安全的基本模型框架，可以描述服务器对于任务的服务以及攻击的响应过程. 其中，模型中的变迁函数可以根据云计算环境中的具体安全问题细化为更加详细的子网，如图 15 中的虚线框内所示，子网可以描述系统对于攻击行为各种层面的阻截和防范机制，将传统的服务过程变为系统的容错或恢复过程. 另外，云计算环境下的租户隔离、网络攻击以及系统自身缺陷等问题，也可以利用不同的子网进行描述，最后形成相应的云安全模型.

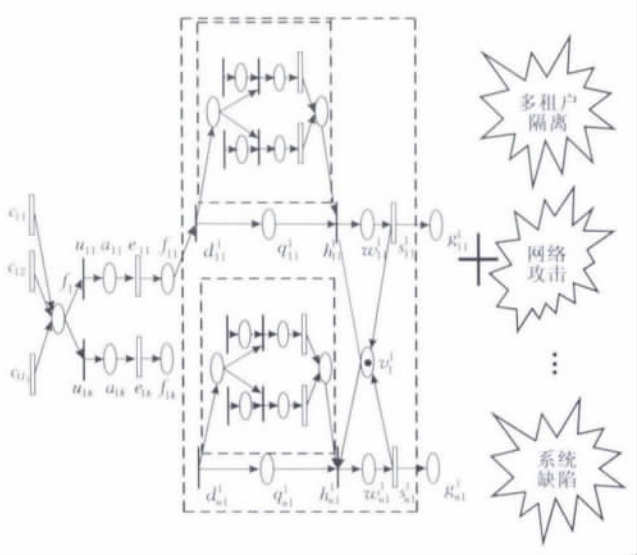


图 15 多队列多服务器云安全模型

对于不同问题的子网安全模型，可以参考已有的 Petri 网可靠性和安全性建模分析研究成果. 文献[74]利用具有很强描述能力的基于马尔可夫再生过程的随机 Petri 网(Markov Regenerative Stochastic Petri Nets，MRSPNS)去建立一个容侵系统的模型，并提出了利用 MRSPNS 描述的系统标准化功能部件模型，该部件模型还可以用于其他容侵系统的安全模型中，如图 16 所示. 图中描述了一个用 MRSPNS 建模的容侵模块，表现了对入侵的容错过程，当一个标记从外部模块进入容侵模块，最终进入检测失败、安全失败等位置的时候，将会向外部模块输出系统失效信号；如果每一个恢复过程都完成了，将会向外部模块输出恢复完成信号. 这个模型可以作为多队列多服务器模型中的一个子网，表示服务器对于攻击性行为的容侵性. Petri 网还可以表示资

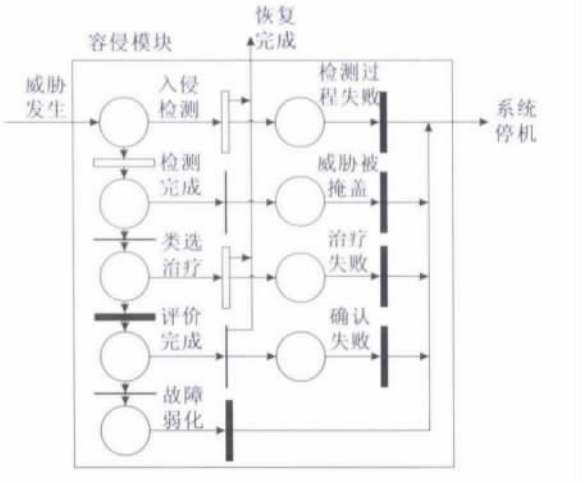


图 16 MRSPNS 的容侵模块[74]

源竞争、共享等情况，着色 Petri 网[75]可以进一步区分多个租户，适用于多租户隔离问题的描述. 另外，文献[76-77]讨论了如何利用 Petri 网进行系统可靠性的建模方法，文献[78]提出了从攻击响应图 ARG 得到相应的 SPN 的方法，从而求解安全的平均失效时间. 文献[79]中利用 Petri 网建立了攻击树模型，并利用仿真工具实现了自动化分析.

## 6 总　结

本文在分析现有云计算服务特性的基础上，深入分析了云计算安全面临的挑战；从云计算架构、机制以及模型评价三个角度，对现有的研究成果进行了研究总结. 基于可信系统设计理念，提出了可管、可控、可度量的云计算安全架构，架构功能的实现需要具体机制以及模型方法予以支持. 文中对安全机制进行了总结分类，分析了云计算环境下安全机制设计的要求，并给出了现有机制的适用范围. 云计算的安全模型在安全保障过程中起着十分重要的作用，文中分类总结了不同指标的形式化描述方法以及不同类型的安全模型. 最后，介绍了一种基于多队列多服务器的云计算安全模型与评价思路. 上述工作比较全面地分析了云计算安全中的重要问题，为进一步的研究工作奠定了基础.

上述研究还存在若干问题需要进一步解决.

云计算安全架构的部署问题：文中提出的安全架构虽然可以很好地保证安全性，但是如何在已有云计算环境上实现增量部署是关键性问题.

云计算安全机制的适用性问题：云计算面临的任务以及环境是不断变化的，这就需要云计算安全机制有着更加灵活的解决方案，覆盖同一安全问题尽量多的意外状况.

云计算安全模型的精确性问题：安全模型规模的增长，使得准确求解异常困难，如何设计较为准确的近似求解方法变得十分重要. 另外，不同的模型方法也各有利弊，如何针对具体问题选择合适的模型方法以提高问题求解的精确度，也需要进一步研究.

## 参　考　文　献


[1] Garg V K. Elements of Distributed Computing. Wiley-IEEE Press, 2002

[2] Foster I, Kesselman C, Tuecke S. The anatomy of the grid: Enabling scalable virtual organizations. International Journal of High Performance Computing Applications, 2001, 15(3): 200-222

[3] Schoder D, Fischbach K. Peer-to-peer prospects. Communications of the ACM, 2003, 46(2): 27-29

[4] Mell P, Grance T. The NIST definition of cloud computing (draft). NIST Special Publication, 2011, 800: 145

[5] Almorsy M, Grundy J, Müller I. An analysis of the cloud computing security problem//Proceedings of the 2010 Asia Pacific Cloud Workshop, Collocated with APSEC2010. Sydney, Australia, 2010: 1-10

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

[7] Ristenpart T, Tromer E, Shacham H, et al. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds//Proceedings of the 16th ACM Conference on Computer and Communications Security. Chicago, USA, 2009: 199-212

[8] Hwang K, Kulkareni S, Hu Y. Cloud security with virtualized defense and reputation-based trust management//Proceedings of the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC'09). Chengdu, China, 2009: 717-722

[9] Kaufman L M. Data security in the world of cloud computing. IEEE Security & Privacy, 2009, 7(4): 61-64

[10] Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, 2011, 34(1): 1-11

[11] Fan Yajun, Liu Jiuwen. Design and implementation of TPM security chip. Information Security and Communications Privacy, 2007(6): 136-137(in Chinese)
(樊亚军，刘久文. TPM 安全芯片设计与实现. 信息安全与通信保密, 2007(6): 136-137)

[12] Garfinkel T, Pfaff B, Chow J, et al. Terra: A virtual machine-based platform for trusted computing. ACM SIGOPS Operating Systems Review, 2003, 37(5): 193-206

[13] Santos N, Gummadi K P, Rodrigues R. Towards trusted cloud computing//Proceedings of the 2009 Conference on Hot Topics in Cloud Computing. Berkeley, USA, 2009: 3-3

[14] Bugiel S, Nürnberger S, Sadeghi A R, et al. Twin clouds: An architecture for secure cloud computing//Proceedings of the Workshop on Cryptography and Security in Clouds (WCSC 2011). Zurich, Switzerland, 2011

[15] Li X Y, Zhou L T, Shi Y, et al. A trusted computing environment model in cloud architecture//Proceedings of the 2010 International Conference on Machine Learning and Cybernetics (ICMLC). Qingdao, China, 2010, 6: 2843-2848

[16] Wailly A, Lacoste M, Debar H. Towards multi-layer autonomic isolation of cloud computing and networking resources//Proceedings of the 2011 Conference on Network and Information Systems Security (SAR-SSI). la Rochelle, France, 2011: 1-9

[17] Bell M. SOA Modeling Patterns for Service Oriented Discovery and Analysis. Wiley, 2009

[18] IBM. IBM Point of View: Security and Cloud Computing. White paper, 2009

[19] Tsai W T, Huang Y, Shao Q. EasySaaS: A SaaS development framework//Proceedings of the 2011 IEEE International Conference on Service-Oriented Computing and Applications (SOCA). UC Irvine, USA, 2011: 1-4

[20] Zhang Min-Jin, Gui Wen-Ming, Su Di-Sheng, et al. The trusted computing technic from terminal to network. Information Technology Letter, 2006, 4(2): 21-34(in Chinese)
(张妥晋, 桂文明, 苏涤生, 等. 从终端到网络的可信计算技术. 信息技术快报, 2006, 4(2): 21-34)

[21] Kallepalli C, Tian J. Measuring and modeling usage and reliability for statistical web testing. IEEE Transactions on Software Engineering, 2001, 27(11): 1023-1036

[22] Halfond W G J, Anand S, Orso A. Precise interface identification to improve testing and analysis of web applications//Proceedings of the 18th International Symposium on Software Testing and Analysis. Chicago, USA, 2009: 285-296

[23] Michael J B, Drusinsky D, Otani T W, et al. Verification and validation for trustworthy software systems. IEEE Software, 2011, 28(6): 86-92

[24] Davila-Nicanor L, Mejia-Alvarez P. Reliability improvement of web-based software applications//Proceedings of the 4th International Conference on Quality Software (QSIC 2004). Braunschweig, Germany, 2004: 180-188

[25] Tsai W T, Zhong P, Balasooriya J, et al. An approach for service composition and testing for cloud computing//Proceedings of the 10th International Symposium on Autonomous Decentralized Systems (ISADS). Chuo-ku, Japan, 2011: 631-636

[26] Gu L, Cheung S C. Constructing and testing privacy-aware services in a cloud computing environment: Challenges and opportunities//Proceedings of the 1st Asia-Pacific Symposium on Internetware. Beijing, China, 2009: 2

[27] Whittaker J. Google V. Microsoft, and the Dev: Test Ratio Debate. 2008

[28] King T M, Ganti A S. Migrating autonomic self-testing to the cloud//Proceedings of the 2010 3rd International Conference on Software Testing, Verification, and Validation Workshops (ICSTW). Paris, France, 2010: 438-443

[29] Zech P. Risk-based security testing in cloud computing environments//Proceedings of the 2011 IEEE 4th International Conference on Software Testing, Verification and Validation (ICST). Berlin, Germany, 2011: 411-414

[30] Li D, Liu C, Wei Q, et al. RBAC-based access control for SaaS systems//Proceedings of the 2010 2nd International Conference on Information Engineering and Computer Science (ICIECS). Zibo, China, 2010: 1-4

[31] Ferraiolo D F, Kuhn D R. Role-based access controls. arXiv preprint arXiv: 0903.2171, 2009

[32] Emig C, Brandt F, Kreuzer S, et al. Identity as a service—Towards a service-oriented identity management architecture//Proceedings of the Dependable and Adaptable Networks and Services. Berlin Heidelberg: Springer, 2007: 1-8

[33] Jiang X, Xu D. VIOLIN: Virtual internetworking on overlay infrastructure//Proceedings of the 2nd International Conference on Parallel and Distributed Processing and Applications. Berlin Heidelberg: Springer, 2005: 937-946

[34] Bari M, Boutaba R, Esteves R, et al. Data center network virtualization: A survey. IEEE Communication Surveys & Tutorials, 2012, 15(2): 909-928

[35] Badger L, Grance T, Patt-Corner R, et al. Cloud computing synopsis and recommendations. NIST Special Publication, 2012, 800: 146

[36] Bowring J, Orso A, Harrold M J. Monitoring deployed software using software tomography. ACM SIGSOFT Software Engineering Notes, 2003, 28(1): 2-9

[37] Ogle D M, Schwan K, Snodgrass R. Application-dependent dynamic monitoring of distributed and parallel systems. IEEE Transactions on Parallel and Distributed Systems, 1993, 4(7): 762-778

[38] Sigelman B H, Barroso L A, Burrows M, et al. Dapper, a large-scale distributed systems tracing infrastructure. Google Research, 2010

[39] Chen M Y, Kiciman E, Fratkin E, et al. Pinpoint: Problem determination in large, dynamic internet services//Proceedings of the International Conference on Dependable Systems and Networks (DSN 2002). Washington, USA, 2002: 595-604

[40] Garfinkel T, Rosenblum M. A virtual machine introspection based architecture for intrusion detection//Proceedings of the Network and Distributed Systems Security Symposium. San Diego, USA, 2003: 16-31

[41] Jones S T, Arpaci-Dusseau A C, Arpaci-Dusseau R H. VMM-based hidden process detection and identification using Lycosid//Proceedings of the 4th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments. Seattle, USA, 2008: 91-100

[42] Jiang X, Wang X. "Out-of-the-box" monitoring of VM-based high-interaction honeypots//Proceedings of the 10th International Conference on Recent Advances in Intrusion Detection. Berlin Heidelberg: Springer, 2007: 198-218

[43] Seshadri A, Luk M, Shi E, et al. Pioneer: Verifying code integrity and enforcing untampered code execution on legacy systems. ACM SIGOPS Operating Systems Review, 2005, 39(5): 1-16

[44] Payne B D, Carbone M, Sharif M, et al. Lares: An architecture for secure active monitoring using virtualization//Proceedings of the IEEE Symposium on Security and Privacy (SP 2008). Oakland, USA, 2008: 233-247

[45] Jones S T, Arpaci-Dusseau A C, Arpaci-Dusseau R H. Antfarm: Tracking processes in a virtual machine environment//Proceedings of the USENIX Annual Technical Conference. Boston, USA, 2006: 1-14

[46] Kruegel C, Kirda E, Mutz D, et al. Automating mimicry attacks using static binary analysis//Proceedings of the 14th Conference on USENIX Security Symposium-Volume 14. Baltimore, Maryland, USA, 2005: 11-11

[47] Hsu F, Chen H, Ristenpart T, et al. Back to the future: A framework for automatic malware removal and system repair//Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06). New Orleans, USA, 2006: 257-268

[48] Suh G E, Lee J W, Zhang D, et al. Secure program execution via dynamic information flow tracking. ACM SIGPLAN Notices, 2004, 39(11): 85-96

[49] Provos N. Improving host security with system call policies//Proceedings of the 12th USENIX Security Symposium. Washington, USA, 2003, 1(8): 10

[50] Goel A, Po K, Farhadi K, et al. The taser intrusion recovery system. ACM SIGOPS Operating Systems Review, 2005, 39(5): 163-176

[51] Prabhakaran V, Arpaci-Dusseau A C, Arpaci-Dusseau R H. Analysis and evolution of journaling file systems//Proceedings of the Annual USENIX Technical Conference. Anaheim, USA, 2005: 105-120

[52] Qin F, Tucek J, Sundaresan J, et al. Rx: treating bugs as allergies—A safe method to survive software failures. ACM SIGOPS Operating Systems Review, 2005, 39(5): 235-248

[53] Srinivasan S M, Kandula S, Andrews C R, et al. Flashback: A lightweight extension for rollback and deterministic replay for software debugging[Ph. D. dissertation]. University of Illinois at Urbana, Champaign, 2004

[54] Bernstein L. Trustworthy software systems. ACM SIGSOFT Software Engineering Notes, 2005, 30(1): 4

[55] Avizienis A, Laprie J C, Randell B, et al. Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, 2004, 1(1): 11-33

[56] Krebs R, Momm C, Kounev S. Metrics and techniques for quantifying performance isolation in cloud environments// Proceedings of the 8th International ACM SIGSOFT Conference on Quality of Software Architectures. Bertinoro, Italy, 2012: 91-100

[57] Catteddu D. Cloud computing: Benefits, risks and recommendations for information security. Web Application Security, 2010: 17-17

[58] Heddaya A, Heldal A. Reliability, availability, dependability and performability: A user-centered view. Computer Science Department, Boston University, 1996

[59] Huang J, Lin C, Kong X, et al. Modeling and analysis of dependability attributes of service computing systems// Proceedings of the 2011 IEEE International Conference on Services Computing (SCC). Washington, USA, 2011: 184-191

[60] Lin Chuang, Wang Yang, Li Quan-Lin. Stochastic modeling and evalu-ation for network security. Chinese Journal of Computers, 2005, 28(12):1943-1956(in Chinese)
(**林闯，汪洋，李泉林. 网络安全的随机模型方法与评价技术. 计算机学报**, 2005, 28(12): 1943-1956)

[61] Brall A, Hagen W, Tran H. Reliability block diagram modeling—Comparisons of three software packages// Proceedngs of the Annual Reliability and Maintainability Symposium (RAMS'07). Orlando, USA, 2007: 119-124

[62] Abd-Allah A. Extending reliability block diagrams to software architectures. System, 1997, 97(80): 93

[63] Leveson N G, Cha S S, Shimeall T J. Safety verification of ada programs using software fault trees. IEEE Software, 1991, 8(4): 48-59

[64] Dugan J B, Sullivan K J, Coppit D. Developing a low-cost high-quality software tool for dynamic fault-tree analysis. IEEE Transactions on Reliability, 2000, 49(1): 49-59

[65] Yager R R. OWA trees and their role in security modeling using attack tree. Information Sciences, 2006, 176(20): 2933-2959

[66] Saini V, Duan Q, Paruchuri V. Threat modeling using attack trees. Journal of Computing Sciences in Colleges, 2008, 23(4): 124-131

[67] Besson F, Jensen T, Métayer D L, et al. Model checking security properties of control flow graphs. Journal of Computer Security, 2001, 9(3): 217-250

[68] Jhala R, Majumdar R. Software model checking. ACM Computing Surveys (CSUR), 2009, 41(4): 21

[69] Martinez J, Muro P, Silva M. Modeling, validation and software implementation of production systems using high level Petri nets//Proceedings of the 1987 IEEE International Conference on Robotics and Automation. Raleigh, NC, USA, 1987, 4: 1180-1185

[70] Hong J E, Bae D H. Software modeling and analysis using a hierarchical object-oriented Petri net. Information Sciences, 2000, 130(1): 133-164

[71] Wang Y, Lin C, Li Q L. Performance analysis of email systems under three types of attacks. Performance Evaluation, 2010, 67(6): 485-499

[72] Madan B B, Goševa-Popstojanova K, Vaidyanathan K, et al. A method for modeling and quantifying the security attributes of intrusion tolerant systems. Performance Evaluation, 2004, 56(1): 167-186

[73] Lin Chuang. Performance analysis of request dispatching and selecting in Web server clusters. Chinese Journal of Computers, 2000, 23(5): 500-508(in Chinese)
(**林闯. Web 服务器集群请求分配和选择的性能分析. 计算机学报**, 2000, 23(5): 500-508)

[74] Fujimoto R, Okamura H, Dohi T. Security evaluation of an intrusion tolerant system with MRSPNs//Proceedings of the International Conference on Availability, Reliability and Security (ARES'09). Fukuoka, Japan, 2009: 427-432

[75] Jensen K. A brief introduction to coloured petri nets//Proceedings of the Tools and Algorithms for the Construction and Analysis of Systems. Berlin Heidelberg: Springer, 1997: 203-208

[76] Malhotra M, Trivedi K S. Dependability modeling using Petri-nets. IEEE Transactions on Reliability, 1995, 44(3): 428-440

[77] Bernardi S, Bobbio A, Donatelli S. Petri nets and dependability//Lectures on Concurrency and Petri Nets. Berlin Heidelberg: Springer, 2004: 125-179

[78] Madan B B, Trivedi K S. Security modeling and quantification of intrusion tolerant systems using attack-response graph. Journal of High Speed Networks, 2004, 13(4): 297-308

[79] Dalton G C, Mills R F, Colombi J M, et al. Analyzing attack trees using generalized stochastic Petri nets//Proceedings of the Information Assurance Workshop. New York, USA, 2006: 116-123

**LIN Chuang**, born in 1948, Ph. D., professor, Ph.D. supervisor. His research interests include computer networks, performance evaluation, network security analysis, and Petri net theory and its applications.

**SU Wen-Bo**, born in 1989, Ph.D. candidate. His research interests include performance evaluation, cloud security and privacy.

**MENG Kun**, born in 1980, Ph. D., assistant professor. His research interests include performance evaluation and stochastic models.

**LIU Qu**, born in 1990, Ph. D. candidate. His research interests include performance evaluation and parallel computing.

**LIU Wei-Dong**, born in 1968, Ph. D., professor, Ph. D. supervisor. His research interests include cloud computing architecture and grid computing.

**Background**

The cloud computing is changing our lives, and its security problem is the most important issue, which people are increasingly concerned with. The security of cloud computing is a hot but difficult topics. In this paper, the authors point out how to design a safe cloud architecture, and provide an overview of the security mechanism and model in the cloud computing. They also propose a security modeling method based on the multiqueue multiserver model. It will be useful for other people's research on relative directions.

This work is partly supported by the National Basic Research Program(973 Program) of China(Nos. 2010CB328105, 2009CB320505), National Natural Science Foundation of China (Nos. 60932003, 61020106002, 61070182, 60973144, 61173008, 61071605). These projects aim to provide better performance and security assurance in computer networks and information systems. The authors has been working on the performance evaluation using the stochastic theoretical models, and now they use these method to evaluate the security of these system. Many papers have been published in respectable international conferences and transactions, such as INFOCOM, IEEE Journal on Selected Areas in Communication and IEEE Transactions on Parallel and Distributed Systems.