云计算环境下隐蔽信道关键问题研究

吴敬征[1,2]，丁丽萍[1]，王永吉[1,3]

（1.中国科学院 软件研究所 基础软件国家工程研究中心，北京 100190；2.中国科学院 研究生院，北京 100049；
3.中国科学院 软件研究所 计算机科学国家重点实验室，北京 100190）

摘 要：首先综述了云计算平台的发展现状、虚拟化关键技术以及云计算安全现状；其次综述了近40年来隐蔽信道分析在操作系统、数据库系统、网络系统领域的发展与研究成果；借助云计算隐蔽信道实例，说明云计算隐蔽信道问题研究的必要性；从理论研究和工程实践角度提出2种新的隐蔽信道分类方式；结合现有研究成果，指出云计算隐蔽信道研究中存在的4个关键问题：① 缺乏云计算环境下隐蔽信道的形式化定义；② 缺乏系统化的标识方法；③ 缺乏威胁度量方法；④ 缺乏相应的安全标准。并明确给出云计算隐蔽信道的形式化定义，该定义可指导隐蔽信道标识、度量的工程实践；最后指出云计算隐蔽信道研究在学术研究、工业生产中的价值和意义。

关键词：云计算；隐蔽信道；虚拟化技术；云安全；隐蔽信道标识；场景构建；信道度量

中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2011)9A-0184-20

Research on key problems of covert channel in cloud computing

WU Jing-zheng[1,2], Ding Li-ping[1], WANG Yong-ji[1,3]

(1.National Engineering Research Center for Fundamental Software, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China;
2.Graduate University, The Chinese Academy of Sciences, Beijing 100049, China;
3.State Key Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

Abstract: First the development of cloud computing, virtual techonolody and the cloud security were surveied. Then the evolvements of the covert channel in operating system, database, and network in the last 40 years were reviewed. Several examples of covert channel were introduced, which indicated the necessity of research. The potential covert channels in cloud computing were classified into two new categories from the aspects of theoretical research and the engineering practice. The four key problems including the lack of definition, the lack of systemic identification and evaluation approach and the lack of security criterions were pointed out. The covert channel in cloud computing was formally defined. Finally, the academic and industrial values of covert channel research are presented.

Key words: cloud computing; covert channel; virtual technology; cloud security; covert channel identification; scenario construction; covert channel evaluation

---

收稿日期：2011-04-11

基金项目：核高基重大专项基金资助项目（2010ZX01036-001-002-2）；中国科学院软件研究所网络算法与数字信息重大基金资助项目（YOCX285056）；计算机科学国家重点实验室自主研究课题（CSZZ0808）

Foundation Items: The National Science and Technology Major Project (2010ZX01036-001-002-2); The Grand Project Network Algorithms and Digital Information of the Institute of Software, Chinese Academy of Sciences (YOCX285056); The State Key Laboratory of Computer Science Funding of Innovative Research of China (CSZZ0808)

1　引言

云计算是一种基于互联网服务的能够弹性动态分配虚拟化资源的新型计算模式，给用户带来全新的计算体验，成为 IT 产业发展的新的经济增长点[1,2]。云计算以创新的计算模式将资源封装成服务，为用户提供了近乎无限的计算能力和信息服务，同时降低了企业购置、维护数据中心等基础设施的投资，提高了生产力，促进了信息产业变革。

典型的云架构分为基础设施层、平台层和应用层，虚拟化技术为其提供了计算资源的可伸缩性、可用性和基于数据隔离保障的安全性[3,4]。数据保护是云计算面临的首要安全问题。如何保障客户数据不被泄漏，是云计算以及虚拟化技术的关键。虚拟化技术固有的隔离性为客户数据提供了一定程度的保护；同时安全人员也提出多种安全策略技术以实现更高强度的访问控制保障。例如，在 Xen[5]虚拟平台中，sHype 实现了 Chinese Wall 和 Type Enforcement 等强制访问控制策略来保证数据的机密性[6~8]；HyperSentry 实现了基于虚拟机监控器的完整性保护机制[9]。然而，即使在安全策略的保障之下，同一硬件平台上的多个虚拟机仍要共享硬件资源，这将会不可避免地导致信息泄漏，而这些信息泄漏的途径就称为隐蔽信道[10]。

隐蔽信道是一个经典的研究课题，早在 1973 年由 Lampson 在程序限制问题的研究中提出[11]。隐蔽信道是指恶意进程通过合谋操作信息系统中的共享资源而实现的一种信息泄漏方式，是传统单机系统、网络操作系统以及数据库系统等的重要威胁[10,12~16]。在云计算平台中，隐蔽信道能够破坏云计算平台的隔离性，泄漏客户的机密信息[17,18]。云计算等新型网络环境下的隐蔽信道问题研究，逐渐成为研究人员近几年关注的焦点[17~21]。

隐蔽信道分析是国内外安全标准脆弱性分析的强制要求，主要包括隐蔽信道标识，即在系统中发现隐蔽信道的存在；隐蔽信道威胁度量，即用容量指标来衡量信道对系统产生的潜在威胁程度；隐蔽信道处置，即从消除、限制和审计角度对信道做处理，以达到消除或限制威胁的目的[22]。在隐蔽信道分析的每一个环节，都存在一些典型的分析方法。例如，信道标识方法包括语法信息流标识算法、语义信息流标识算法、共享资源矩阵法、无干扰分析方法、隐蔽流树分析法等；信道度量方法包括形式化的推导算法和非形式化的计算方法等；信道处置方法包括添加噪音和干扰等方法和基于多概率的隐蔽信道检测方法等[10,22~24]。

1973 年到 1986 年间，隐蔽信道分析的对象聚焦在操作系统上，侧重于从形式化顶层规范、描述性顶层规范、系统参考手册以及系统源代码角度进行标识，发现潜在的隐蔽信道[25]；1987 年以后研究人员更加侧重网络隐蔽信道的研究，从网络协议到具体的网络场景，发现了大量的网络存储和时间隐蔽信道；2000 年以后对网络隐蔽信道的研究更加深入，从隐蔽信息的编解码机制、传输反馈机制、检测审计、消除限制等机制涌现出大量的实用方法[16,26~28]。

云计算兴起之后，隐蔽信道作为威胁数据安全的关键问题，又一次引起研究人员的重视[17,18,20]。Aviram 等研究人员从隐蔽信道的角度总结了云计算面临的 4 类安全风险[20]：首先，相比单处理器或单线程的处理器，云计算平台拥有大量的并行操作，而其精确的时钟则可以用来作为时间隐蔽信道的共享媒介，实现基于时间参数的信息泄漏；其次，攻击方式从系统内部转移到外部，恶意进程通过观察 CPU 负载、缓存响应时间等信息推测部署在同一硬件平台上的其他虚拟机的有效信息，从而进一步窃取机密信息；再次，由于隐私保护和商业规则，云服务商无法记录和监控客户执行的操作，导致信息泄漏、隐蔽信道等攻击方式难以记录和发现；最后，现有的信息泄漏、隐蔽信道处理技术不能直接照搬到云计算环境中，例如，资源分区、添加噪音和延迟等机制虽然能够限制隐蔽信道的威胁，但是这些措施同样会影响云平台的效率和弹性，从而降低其服务质量。云计算受到产业界、学术界和政府各界越来越多的关注，对其安全性能的要求也呈现逐渐上升的趋势。要解决以上问题，就必须对云计算环境下的安全问题，尤其是隐蔽信道的机理深入研究，才能够为云计算发展提供更安全的环境。

本文立足于云计算平台，力求简明扼要的梳理隐蔽信道的研究历史，指明在新型计算平台中隐蔽信道研究面临的主要问题以及今后的发展方向、解决方案及技术发展趋势。并试图为云计算隐蔽信道研究方向勾勒出一个较为全面和清晰的概况，为隐蔽信道相关领域的研究者提供有益参考，推动云计算安全分析、尤其是隐蔽信道分析的标准化进程。

2 云计算及其安全现状

文献[29]指出网络是群体的象征，无数的个体思维聚在一起，形成了无可逆转的社会性。它所表达的既包含了计算机的逻辑，又包含了大自然的逻辑，进而展现出一种超越理解能力的力量。

孤立的计算机作为计算工具只能提供有限的计算能力和资源，而一旦计算机相互连接将会发生颠覆性的变化。云计算基于互联网服务，以全新的计算模式为用户提供了近乎无限的计算能力和信息服务。

云计算包罗万象，从逻辑上抽象了计算机网络的软硬件资源，是当前学术界和工业界的研究热点。为了在云计算的链条中占有一席之地，研究人员从各自熟悉的领域分别给出云计算的定义，典型代表如下。

IBM 公司认为，云计算是用来同时描述一个系统平台或者一种类型的应用程序，云计算平台按需动态地部署、配置、重新配置以及取消服务。在云计算平台中的服务器可以是物理的服务器或者虚拟的服务器。高级的计算云通常包含一些其他的计算资源，例如存储区域网络(SAN)、网络设备、防火墙以及其他安全设备等[2,30]。

Berkeley 大学的研究人员认为，云计算指在互联网上以服务形式提供的应用，也指在数据中心中提供这些服务的硬件和软件，而这些数据中心中的硬件和软件则被称为云[31,32]。

美国标准与技术研究院(NIST, national institute of standards and technology)概括了云计算的5大特点、3大服务模式和4大部署模式，认为云计算是一种按使用量付费的服务模式，这种模式提供可用的、便捷的、按需的网络访问，使用可配置的计算资源共享池(包括网络、服务器、存储、应用软件和服务)，这些资源只需投入很少的管理工作，或与服务供应商进行很少的交互就能够快速地提供给客户[4]。

虽然云计算的定义各有侧重，但从本质上讲，云计算是计算机硬件技术、软件技术和网络应用发展的必然产物。云计算与传统的普适计算、网格计算的区别在于，云计算提供了一种全新的商业模式和用户体验，用以人为本的观点为客户提供了经济易用的服务。因此本文作者认为，云计算是一种基于互联网服务的、能够弹性动态分配虚拟化资源的新型计算模式。云计算模式中的参与对象包括云服务商和用户。云服务商根据用户的需求，安装、配置、部署计算资源，将软硬件封装成服务的形式提供给用户；用户可以使用诸如智能手机、平板电脑、智能家电等瘦终端通过互联网使用定制的服务，对于用户而言，云计算服务是透明的，像公共的水和电资源一样，随时方便使用；而在特定情况下，用户也可以成为更高级用户的云服务商，例如使用基础设施云的应用云提供商。为了更清楚地分析云计算的实际应用及特征，必须要了解云计算架构、关键技术以及为了保障云计算的可用性而采取的安全措施等。

2.1 云计算架构

NIST 概括了云计算的3种服务模式，按照从硬件到应用的层次，依次分为基础设施即服务(IaaS, infrastructure as a service，)、平台即服务(PaaS, platform as a service)以及软件即服务(SaaS, software as a service)[1,4,31,32]，如图1所示。

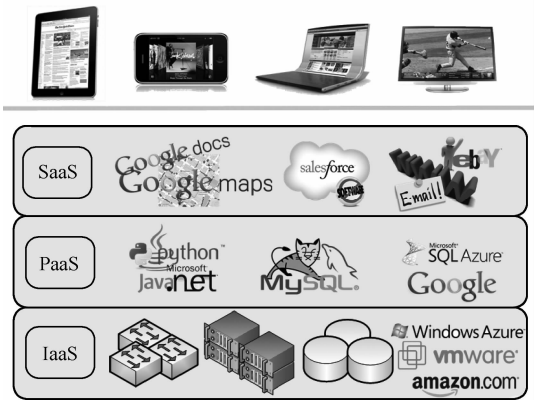

图1 云计算架构示意图

基础设施服务商以数据中心为基础，为客户动态地提供计算资源、存储资源和网络资源。使用基础设施云，客户企业免去了购买、安装、配置和维护数据中心的环节，降低了经营成本，提高了运营效率。Amazon EC2[33]是典型的基础设施服务平台，通过Xen虚拟化技术，为用户提供虚拟硬件资源。用户按照需求通过 EC2 的控制界面定制生成云计算虚拟机实例，并部署自己的平台系统直至一个完整的生命周期结束。客户只需为自己使用的计算资源付费，节省了设备购买与维护费用。Windows Azure 和 Vmware 都提供了基于计算机硬件、操作系统和应用资源的云服务模式，可有效地控制硬件资源、降低运营成本。

平台服务商为用户提供了丰富的“云中间件”资源，包括面向开发人员的数据库逻辑、Web 应用逻辑和编程开发环境等。Google App Engine[34]是典型的部署在基础设施云上的网络应用程序。App Engine 应用程序易于构建和维护，并可根据访问量和数据存储需要的增长轻松扩展。Google App Engine 支持 Java、Python、•Net 等编程语言。在 IaaS 中，客户只需为使用的资源付费，可以控制应用程序消费的最大资源量，使其一直保持在预算范围内。IaaS 为平台上托管的应用提供了良好的自动伸缩性和高可用性。

软件服务商交付给客户的是定制的软件应用，以服务的方式租赁给客户。客户无需购买软件，只需“按需付费”。云计算 SaaS 软件提供商将软件以服务的方式部署在云平台中，方便软件的发布与升级。软件服务的变革提高了用户的体验，节省了开发部署成本。Salesforce.com [35]的 CRM（客户关系管理）软件、Google 公司的 Gmail 和 Docs 等，是软件即服务的典型代表。

云计算凭借对服务和计算资源的高效管理和弹性分配，充分体现了“网络就是计算机”的思想。在云计算架构中，大部分运算交由云来完成。对于云用户来说，可以使用性能不强的智能手机、智能家电、平板电脑等瘦终端设备获取云计算服务(如图 1 所示)。瘦终端具有性能稳定、故障率低、安全性高、绿色环保等特点，有效降低了终端用户的系统拥有成本。

2.2　虚拟化技术

云计算的基础设施层、平台层和软件应用层的每一层都为上层提供了透明的计算资源，而虚拟化技术是实现计算资源抽象的核心技术。虚拟化是资源的逻辑表示，不受物理限制的约束。

虚拟化技术通过在硬件平台上添加一层薄的虚拟机监控 (VMM，virtual machine monitor) 程序，实现对处理器、内存管理器(MMU)和 I/O 系统等的虚拟化管理。虚拟机监控程序又称为监控程序(hypervisor)[5]。从应用层来看，程序运行在虚拟机上如同运行在实体计算机上一样。VMM 上可以运行多个虚拟机(VM，virtual machine) 程序。VMM 向上层的每个 VM 提供原始的硬件接口。每个 VM 针对不同的应用可运行相应的操作系统和应用程序，VM 之间不会相互影响。

虚拟机直接管理硬件资源，为上层应用提供透明的接口，广义上可以认为是一种新型的操作系统，与传统的操作系统相比，虚拟机具有以下显著的特点：

1) 隔离性。在云计算平台中，云服务商可能同时为竞争公司提供服务，保障数据不被泄漏对客户而言至关重要。虚拟化技术在 VM 之间提供了强隔离机制，配合强制访问控制策略，能够保障客户数据安全。虚拟机可以根据需要运行不同的操作系统和应用程序，即使某个虚拟机内部运行的程序崩溃也不会影响到其他客户数据。

2) 封装性、弹性。基于虚拟化技术，每个完整的 VM 都可以看成独立的逻辑实体，内含操作系统、存储系统和应用程序，便于系统的备份、复制和迁移。基于虚拟化的封装性，VM 可以在不同的物理硬件上热迁移，平滑过渡且用户无需察觉。封装性对于云客户的动态资源调整和计算资源弹性分配至关重要，同时也为数字版权分发带来新的思路。

3) 高效性。虚拟化技术已经发展了近半个世纪，即使没有硬件虚拟化的支持其效率几乎和物理机器差距不大。近年来芯片厂商提供了硬件虚拟化支持，将 VMM 功能逐渐转移到电路级的硬件中，大大提高了虚拟机的效率。在软件层上，每个 VM 中可以只运行单独的程序和为该程序定制操作系统，给应用程序提供了更专用的硬件和操作系统资源，相比复杂的多程序并行的系统平台，进一步提高了运行效率。

虚拟化技术为创建基于网络服务、动态资源分配的云计算平台提供了完整的技术支撑。虚拟化技术为企业节约了成本，同时极大地提高了计算资源的利用率，成为业界最受关注的热点技术之一。

Citrix 公司的虚拟化解决方案基于 Xen 体系结构实现。Xen 首次提出全虚拟化和半虚拟化的概念，为不同类型的 CPU 架构提供了强大的虚拟化支持，Amazon EC2 平台就基于 Xen 系统平台。Vmware 公司拥有 3 条虚拟化产品线：数据中心产品、桌面产品和其他虚拟化辅助产品。涵盖了服务器虚拟化的整个生命周期，为客户实现虚拟化基础设施、整合资源，提高资源利用率，在降低运营维护成本的同时，增强业务的灵活性、可用性和安全性。Microsoft 公司通过 Window Server 2008 和 Hyper-V 进入服务器虚拟化市场，提出从“数据中心到桌面”的虚拟化战略。IBM 公司提出“虚拟一切资源”的战略，整合现有的虚拟化技术，在更

高层次上实现计算资源和存储资源的虚拟化，在虚拟化数据中心内实现跨虚拟化平台智能的、动态的资源调度[1]。

2.3 云计算安全

云计算将基础设施、平台及应用部署到云端，把用户从繁重且昂贵的运营与维护中解脱出来，允许客户仅使用瘦客户端消费云服务。针对这种新的服务模式，无论是从观念上还是技术上都对安全性能带来极大的挑战。对于入侵者而言，云计算平台提供了一个廉价、高效、稳定易用的入侵平台；而客户则担心将应用程序与服务部署在不可控的环境中的安全性。从云服务的生命周期来看，云计算的主要面临以下几个重要的安全威胁[36~39]，如图2所示。

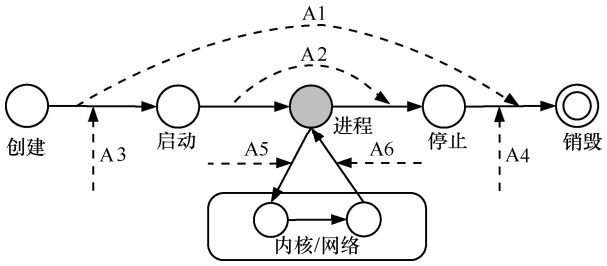

图2 虚拟机生命周期中面临的安全威胁

1）启动与停止阶段(A3，A4)。云计算的易用性允许恶意用户在短时间内启动和部署大量的计算节点用于恶意攻击，例如用于DDoS、Botnet[40]等攻击方式或者对于机密信息的暴力破解[36]。图2中A3表示篡改启动镜像类型的攻击，恶意用户篡改替换VM启动的镜像文件，导致客户在云服务的启动阶段就已经被植入恶意程序，成为入侵者的攻击对象；A4表示篡改持久化存储的虚拟机攻击方式，当虚拟机将客户数据写入到持久化设备中后，恶意程序将客户信息泄露给攻击者或者造成客户数据的故意丢失。云计算导致客户的攻击从操作系统内部转向到操作系统外部，A3和A4的攻击类型都是针对云平台新的攻击方式，可以采用完整性策略对系统进行安全防范。

2）虚拟机运行阶段(A1，A2)。虚拟化技术是云计算平台的核心，为虚拟机中的操作系统和服务提供了硬件层的逻辑抽象。为了保证应用层服务能够相对平等高效的共享底层硬件，虚拟化技术提供了大量的共享资源，而这些共享资源则成为隐蔽信道发生的源泉[20]。A1表示虚拟机之间基于共享资源的隐蔽信道，例如，基于CPU负载和Cache缓存的隐蔽信道[17,18]。在云计算平台中，虽然VMM为每个虚拟机分配了虚拟CPU，但是最终的任务仍然要顺序地在物理CPU上执行，通过观察物理CPU的负载状况，能够推测同一物理平台上其他虚拟机内的机密信息；基于Cache缓存的隐蔽信道类似于CPU负载信道，通过访问Cache的延迟时间，泄漏虚拟机的机密信息。A2表示虚拟机内部的隐蔽信道，例如，针对Linux操作系统的事件标识型隐蔽信道[13]，该信道的收发双方通过改变和观察特定事件的状态，合谋传递机密信息。A1和A2分别表示了虚拟机外部和内部的2种信息泄漏方式，这2种方式在虚拟机和操作系统层都是不可避免的，即使部署了强制访问控制策略，仍然无法彻底清除隐蔽信道[37]。

3）应用程序运行阶段(A5，A6)。对终端用户而言，云计算的服务最终由应用程序提供，恶意软件和风险程序是其重要威胁。A5表示木马或者病毒攻击方式，当执行内核的系统调用时，木马程序劫持系统调用并执行恶意操作破坏系统；A6表示返回值篡改攻击，木马程序劫持程序并返回错误的结果从而破坏系统安全，例如缓冲区溢出攻击等。在网络环境中，A5和A6表示中间人攻击方式和其他的网络攻击方式，劫持网络会话执行恶意操作。在云计算环境下，基于Web 2.0的攻击方式和基于浏览器的信息泄漏方式都对系统造成重要威胁[41,42]。应用层攻击处在虚拟机内部，并不是云计算的新型产物，涵盖了传统的攻击方式。如何防护和限制恶意程序的破坏性，保护数据和系统的安全，是相对成熟的研究领域。

这3种类型的安全威胁覆盖了云服务完整的生命周期。按照由低向高的层次，可视为针对VMM、VM和应用程序的攻击。前2种攻击方式利用了云计算平台动态易用、资源共享的特点，是安全研究领域的新问题。基于应用程序的攻击虽然是传统的研究领域，但是随着计算机技术的飞速发展，攻击方式和攻击手段不断创新，消除和防范技术愈发复杂。如何从根本上消除这些安全威胁，是目前安全领域研究的重点，目前常用的技术包括在虚拟机平台层部署安全策略，如sHype[6,7]、HyperSentry[9]、HyperSafe[43]、Lares[38]等机制分别实现了包含数据机密性和完整性的强制访问控制策略。通过安全策略，配置私有云、公有云和混合云以创建灵活实用的云平台。然尔，即使部署了安全策略，只要存在

硬件资源共享，就不可避免地产生隐蔽信道导致信息泄漏。由于隐蔽信道的不可彻底消除性，只能深入研究其机理，尽可能地采取消除、限制、审计和检测等措施。

3　隐蔽信道研究现状

隐蔽信道是指恶意进程通过合谋操作信息系统共享资源而实现的一种信息泄漏方式。隐蔽信道分析是国内外安全标准(TCSEC[44]、CC[45]、GB17859-2001[46]、GB/T20272-2006[47]等)脆弱性分析的强制要求，分析工作包括信道识别、度量和处置[10]。

信道识别是对系统的静态分析，强调对设计和代码进行分析以发现所有潜在的隐蔽信道；信道度量是对信道传输能力和威胁程度的评价；信道处置措施包括信道消除、限制、审计和检测，隐蔽信道消除措施包括修改系统、排除产生隐蔽信道的源头、破坏信道的存在条件；限制措施要求将信道危害降低到系统能够容忍的范围内；并非所有的潜在隐蔽信道都能被入侵者实际利用，如果对所有的潜在隐蔽信道进行度量和处置会产生不必要的性能消耗、降低系统效率，隐蔽信道检测强调对潜在隐蔽信道的相关操作进行监测和记录，通过分析记录，检测出入侵者对信道的实际使用操作，为信道度量和处置提供依据。

隐蔽信道问题自提出到现在的近 40 年时间中，研究对象范围涉及程序限制问题、单机操作系统、数据库系统和网络操作系统，以及当前流行的云计算平台。从理论研究到工程实践，在隐蔽信道分析的每一个环节，都存在一些实用且具有指导性的分析方法[10,16,22,23]。然而由于隐蔽信道在不同系统中呈现出的多样性和复杂性，这些方法各有侧重，并没有一种通用的方法从根本上解决隐蔽信道的所有问题。例如，在信道标识过程中语法信息流存在大量的误报，共享资源矩阵方法容易产生状态爆炸；在信道度量过程中，单纯的使用容量指标并不能全面度量信道威胁；在处置过程中，隐蔽信道难以全面消除且只能采用限制和审计的方法，而在大量的系统正常操作中检测隐蔽信道容易产生误报和漏报。如何解决这些问题，仍然是当前隐蔽信道问题研究的重点。

隐蔽通道分析是信息安全研究领域的重要难题，其原因主要在于：首先，隐蔽通道分析是一个应用范围狭小的高端技术，除了从事高安全等级信息系统研究的人员以外，大部分研究者很少有机会接触到隐蔽通道分析问题；其次，隐蔽通道分析建立在信息系统强制访问控制等安全机制的研究基础上，研究起点比较高；再次，分析千万行的源代码，工程实现复杂；而且由于发达国家的技术壁垒，有用的参考资料非常少。隐蔽信道分析关系到国家信息系统安全的战略需求，因此必须从机理上深入研究并加以防范和治理。

3.1　隐蔽信道分析对象

隐蔽信道广泛存在于高等级安全信息系统中，与系统的强制访问控制策略模型相关。在安全操作系统和安全数据库系统中，高安全级和低安全级用户之间通过修改和感知共享变量的值或者属性传递信息，泄漏机密信息。网络信道包括多级安全网络传输信道和普通网络传输信道。其区别在于网络环境中是否部署了多级安全策略。普通网络信道不涉及安全级别，该信道期望在通信链路上附加一层隐蔽通信，目前对其的研究逐渐占据了主流地位。

3.1.1　操作系统

操作系统隐蔽信道研究重点在于防患于未然，侧重于信道标识、场景构建、容量度量和信道处置。隐蔽信道标识对操作系统进行分析，切入点包括用户手册、形式化顶层规范、详细顶层规范以及系统源代码。标识方法的设计源于对隐蔽信道概念的理解，研究人员从工程实践的角度给出隐蔽信道多种不同的定义，由此产生了诸如语义/语法信息流方法、共享资源矩阵法和无干扰分析法等信道标识方法[10,23]。

Kemmerer[48~50]认为隐蔽信道是使用非正常数据客体的项从一个主体向另一个主体传递信息的信道，并由该定义设计出共享资源矩阵法，该方法曾成功应用于几个项目(如 Unix 2、DG/UX 等)。Kemmerer 指出，隐蔽信道的存在归根于系统中的共享资源，如果能够找出所有用于读/写的系统资源及其上操作并对这些资源进行分析就能够找到相应的隐蔽信道。但是从源代码层次构建共享资源矩阵工作量巨大，面对千万行的大型系统时，容易产生状态爆炸，从而使标识工作难以继续。

Tsai[51]等人认为隐蔽信道是违反强制安全策略模型的 2 个主体间的非法通信。并由此提出语义信息流方法，该方法分析编程语言的语义、内核代码中的数据结构，发现其中变量的可修改性和可见

性；然后利用信息流分析方法来判断内核变量的间接可见性，从而发现信道的收发进程和共享资源以此发现潜在的隐蔽信道。但是该方法仍然存在着工作量大、缺少自动化工具的缺点。

Tsai的方法源于Denning[52]的语法信息流标识方法。Denning首先对信息流模型进行了形式化描述，并确立了安全级之间信息流的偏序关系。该方法首先将语句抽象为包含“明流”和“暗流”的信息流，然后将信息流策略应用于系统的顶层规范或者代码上，生成信息流公式，最后利用定理证明器证明信息流公式的正确性。如果信息流公式不能被证明，则可能存在隐蔽信道。该方法搜索彻底，但是可能会产生大量的伪非法流，增加了人工分析的负担，并且难以找出放置隐蔽信道处置代码的具体位置。

卿斯汉[22,53]延续了语义信息流的思想，设计了一种代码层次的标识方法称为回溯搜索法，该方法引入“剪枝规则”，在标识过程中立即删除不能构成隐蔽信道的共享变量，显著地减少了分析的工作量。该方法被应用于安胜OS v4.0系统的隐蔽信道识别中，成功地发现了18条真实的隐蔽信道。

Goguen[54]认为在安全系统中一个用户不能意识到任何不由它所支配的用户的任何操作，称为无干扰模型。进程之间无干扰时具有以下性质：如果进程A的输入不能影响进程B的输出，则不可能从进程A向进程B传输信息。因此，如果多级安全系统中不存在隐蔽信道，则任何一个用户都应该与其支配的用户之间满足无干扰关系。该方法可以增量进行，但该方法是一种乐观的方法，不适合分析大规模系统。

研究现有的方法可知系统源代码囊括了系统所有的信息，是所有潜在隐蔽信道的藏身处，从而导致隐蔽信道标识的难点在于如何在海量的系统源代码中查找出符合隐蔽通信特征的信道。因此，抽象隐蔽信道特征，可以将其形式化地表述为$<V,PA_h,PV_l,P>$[10,13]，其中V代表共享资源，PA_h代表能够修改共享资源V的高安全级主体，PV_l代表能够观测V值发生变化的低安全级主体，P代表安全信息系统的非自主访问控制策略，并且PA_h的安全级支配PV_l的安全级，表示为$PA_h \succ_P PV_l$。在安全策略P的保障下，不允许信息流从PA_h流向PV_l。隐蔽信道就是系统中所有违反安全策略P，导致信息流从PA_h流向PA_h的通信信道。对应于系统源代码，共享资源V表示系统共享变量；PA_h和PV_l表示不同的用户进程，且对共享资源V的修改始终满足V的值域，即$V\ \{PA_h\ \ PV_l\}^{P}\ V$。在安全策略$P$的保障下，不允许信息流从$PA_h$流向$PV_l$。隐蔽信道标识就是发现系统中所有违反安全策略$P$，导致信息流从$PA_h$流向$PV_l$的潜在隐蔽信道。

根据定义$<V,PA_h,PV_l,P>$，参考Denning的语法信息流方法，文献[25]提出了一种针对操作系统源代码的基于有向信息流图的隐蔽信道标识方法。对于大型系统，该方法首先按照高内聚低耦合的原则将系统划分成相对独立的子模块，然后对具体模块单独分析，模块间的耦合再做后续分析。对于具体的模块，使用LLVM[55]编译工具将其编译成等价的更具结构化的中间代码，然后设计了一种针对此中间代码的搜索方法，查找能够同时被修改和访问的共享资源。并为此共享资源及其操作分支进程创建有向信息流图，如果该图中信息流分支是内核外部可见的(即用户态可操作的)，则认为发现了一条潜在的隐蔽信道。对于发现的潜在隐蔽信道，要在部署了安全策略的实验平台上为其构建场景，如果机密信息能够被传输，则该潜在信道为真实隐蔽信道，否则为误报。在对Linux kernel2.6.18内核主要的130万行代码的分析中，使用该方法标识出了40多条隐蔽信道，并根据信道特征划分成4类典型类型进行场景分析[13]，证明了绝大部分为真实的隐蔽信道，且大部分为首次发现。

3.1.2 数据库系统

安全系统通常使用多级安全策略保证系统的机密性和完整性，同时采用支持优先级的并发控制协议解决不同安全级用户的并发访问冲突。即使在安全和实时策略的共同限制下，恶意用户仍然可以利用不同安全级事务之间并发冲突的场景来构造隐蔽信道，传输机密信息。在数据库系统中，隐蔽信道研究主要集中在信道检测、威胁度量和限制技术中。数据库系统中的隐蔽信道主要包括以下3类[56]：

1) 数据库存储资源引入的信道。该信道利用数据库中的共享资源构建信道，如数据、数据字典等。在该隐蔽信道具体场景中，发送者修改数据/数据字典，接收者则通过完整性约束等方式间接感知数据/数据字典的修改，以此来传输机密信息。

2) 数据库管理资源引入的信道。数据库系统中的另一类共享资源包括系统变量、游标、临时数据区等。收发双方通过耗尽有限的共享资源传输机密

信息。另外，同时管理多个安全级别用户的系统安全机制也可能引入信道，如审计机制等。

3) 事务并发控制引起的隐蔽信道[14,15,56]。安全数据库系统中通常依据 BLP[57]模型实施强制访问控制，约束用户的数据访问操作，以保证数据的安全性。同时，为了保证数据操作的实时性，系统还需要采用实时算法处理事务调度和并发控制，如 2PL-Priority[58]协议等。入侵者可以利用不同安全级事务间的并发冲突构造隐蔽信道，称作数据冲突隐蔽信道(data conflict covert channel)[59]。

数据冲突隐蔽信道依赖于不同安全级别事务间的并发冲突，入侵者通过对冲突发起时间的精心设计和编码传递机密信息，属于时间隐蔽信道。数据冲突隐蔽信道中，如果 2 个不同安全级别的用户发起的事务 tr_l 和 tr_h 共同访问同一数据项 dx，且其安全级别关系为 $SL(tr_h) \geqslant SL(dx) \geqslant SL(tr_l)$。其中，低安全级别事务 tr_l 写访问 dx；高安全级别事务 tr_h 读访问 dx。在该场景下，可构造多种具体的数据冲突隐蔽信道，实现高安全级用户向低级别用户传递信息。入侵者对并发冲突的蓄意控制会影响其时间特征，造成冲突间隔时间的随机性减弱规律性增强，因此可以通过监测冲突间隔时间的规律性特征，判断是否发生数据冲突隐蔽信道。

3.1.3 网络系统

网络隐蔽信道将信息泄漏威胁从系统内部转移到系统之间。在网络隐蔽信道中，攻击者期望在最短的时间内准确地传输更多的机密信息；系统安全人员则期望能够检测到机密信息的传输过程，采取相应的处理措施，限制或消除隐蔽信道的威胁。因此，网络隐蔽信道的研究重点在于如何提高隐蔽信道的传输效率、传输准确性以及如何准确检测隐蔽信道和限制其威胁[10,16,60]。

网络隐蔽信道的具体实现技术包括[16]：在数据分组的数据字段隐藏信息[61]、在数据分组的包头字段隐藏信息[62,63]或者利用数据分组的时间属性隐藏信息[64,65]。第一种方式属于信息隐藏的范畴，主要研究嵌入信息的鲁棒性，包括隐写术和数字水印等技术[61,66]；后两者分别称为网络存储隐蔽信道和网络时间隐蔽信道，网络存储隐蔽信道一般将信息附加在不常用的数据段中，包括未用的 IP 头字段(ToS 字段、DF 和 URG 位)、IP 头的扩展和填充段、IP 标识和碎片偏移等[63]，网络时间隐蔽信道将机密信息编码成数据分组的发送/到达时刻、时间间隔等序列[64,67~70]，更加难以检测和处置。

1987 年，Girling[71]发现了 3 种局域网上的隐蔽信道，开启了网络隐蔽信道研究的先河；1996 年 Handel[62]对 OSI 网络模型进行了深入分析，指出理论上存在的隐蔽信道；随后 Rowland[72]发现了 TCP/IP 协议中存在的隐蔽信道实例；Zander[16]综述了 1987 年到 2006 年间网络隐蔽信道的设计实现以及相应的检测与限制技术成果。网络信道的威胁得到了广泛的认可，网络隐蔽信道的检测、度量和处置逐渐成为隐蔽信道研究领域的热点[12,73]。

2004 年美国 Purdue 大学的 Cabuk[65]提出了一种 IP 时间隐蔽信道，称作 IPCTC (IP covert timing channel)。在 IPCTC 中，收发双方约定使用 ACSII 码对隐蔽信息编码，在固定的时间段内，发送方发送一个数据分组，或者保持静默，分别代表符号 1 和 0，构成二元隐蔽信道。当发送方希望发出符号 1 时，其将在时间段内发送数据分组；当希望发出符号 0 时，则不发送数据分组。在固定时间段内，接收者如果接收到该数据分组，则识别符号 1，否则识别符号 0。在 RTT=31.5ms 的网络环境中，固定时间间隔取 60ms，信道容量达到 16.67bit/s，解码错误率为 2%。IPCTC 是简单的二元隐蔽信道，收发双方需要同步机制才能保证正确通信。IPCTC 的问题在于信道容量较小，且当网络中存在抖动和延迟时，可能导致传输错误。

2009 年上海交通大学的 Yao[69]将 IPCTC 信道归纳为 OTC 信道(on/off timing channel)，研究数据分组间隔时间与解码错误率之间的关系，并将解码过程建模为状态转换模型，从信息论的角度推导出 OTC 信道容量计算公式。虽然在数据分组发送间隔为 4.5ms 的情况下，容量达到最优值 180.49bit/s，但 OTC 本质上是二元信道，容量增加的原因在于发送间隔的缩小。IPCTC 中的传输错误问题，在 OTC 中依然存在。2010 年 Zi[73]等人对 Yao 的研究做了一定的改进。

与 IPCTC 信道类似，2005 年美国 Dartmouth 学院的 Berk[67]提出了另一种基于数据分组间隔的时间隐蔽信道。发送端使用单位间隔时间 DT 表示符号 0，2DT 表示符号 1。接收端从接收到第一个数据分组开始计时，当收到下一个数据分组时，利用 2 个数据分组的间隔时间来判断发送的数据。收发双方根据预先约定的时间间隔编解码，构成自同步信道，不需要额外的同步机制。Berk 只研究了 ASCII 编码

的情况，但该方法可以扩展到多元编码，例如使用(k+1)DT表示符号k，进而提高隐蔽信道的容量。

2009年Purdue大学的Sellke[74]中提出了一种基于编码表的网络时间隐蔽信道，称为L bit to n packets信道。发送方使用n个连续的时间间隔$T_1,T_2,\cdots,T_n$发送L bit字符串；接收方在字母表中查找收到的n个时间间隔$R_1,R_2,\cdots,R_n$，并解码成对应的L bit字符串，从而实现隐蔽信息传输。Sellke详细分析了L和n的选择方案，以及在不同网络环境中L和n的取值对隐蔽信道容量的影响。当该信道取值为9bit to 3 packets时，隐蔽信道容量为37bit/s，编码表中包括2^9条数据项。在信道传输中，即使解码过程发生错误，也只有L bit字符受影响。相比IPCTC，该信道提高了信道容量、降低了解码错误率。

分析现有的网络时间隐蔽信道研究成果可以发现，改进信道的编解码过程能够提高信道容量。基于编码表的隐蔽信道虽然在收发双方增加了编解码工作量，但是整体上压缩了传输数据量，提高了信道容量。因此文献[12]在网络时间隐蔽信道的研究中引入了基于Huffman编码的机制，利用敏感信息的编码冗余，降低传输信息量、缩短传输时间，从而提高信道容量。同时Huffman编码是唯一码即时码，一旦传输过程中出现错误，能够及时发现。如果配合纠错协议，能够大幅降低编码错误率。

3.2 关键技术

网络隐蔽信道原型如图3所示，主机A和B分别为隐蔽信道的发送和接收方。在传输周期启动之前，主机A B约定同步周期以及编码表。对于要发送的机密信息，A首先依据编码表对其编码，转换为相应的共享资源操作序列，如数据分组间隔时间、或者ToS字段等；然后发送方A修改共享资源属性；接收方B根据事先约定的传输周期，获得共享资源属性系列，再查找编码表解码得到该隐蔽信息[12]。

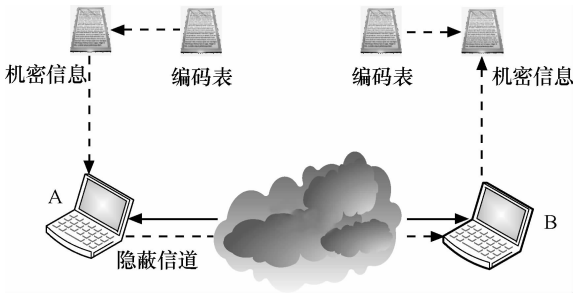

图3 网络时间隐蔽信道原型

图3是隐蔽信道传输的抽象模型，当A B为操作系统内符合安全策略规范的进程时，被修改和访问的共享资源为操作系统中的共享变量或者其他共享状态属性，则该模型为操作系统隐蔽信道模型；当A B为安全实时数据库中不同安全级的用户时，该模型为数据库隐蔽信道模型。

任何类型的隐蔽信道使用者都关注隐蔽信道的传输效率、传输准确率和传输隐匿性，攻击者以容量、准确率和隐匿性指标为目标改进传输机制和编解码机制，以实现更加高效且难以检测的信道；安全人员以这3个指标为指导，设计更加高效的检测、限制和消除方法。因此，隐蔽信道传输机制和编解码机制是影响隐蔽信道效率的关键技术[12]。

3.2.1 传输机制

隐蔽信道本质上是信息传输信道，传输机制的重点集中在对传输介质的研究。根据共享资源属性的不同，传输介质分为存储类型和时间类型，由此衍生出存储隐蔽信道和时间隐蔽信道的分类。

在操作系统中，由于系统资源处在独立系统内部，对于存储的属性或者状态在一定程度上能够被大多数进程共享，因此极易产生存储隐蔽信道；存储隐蔽信道属于静态资源，能够被修改和访问。而进程间的相互影响会产生时间上的冲突由此带来时间隐蔽信道。相比存储隐蔽信道，时间隐蔽信道体现了更多的动态特征，因此更加难以检测和控制。

在数据库系统中，持久化的数据资源更加丰富，因此可能产生更多的存储隐蔽信道；由于存在大量用户的并行操作，时间隐蔽信道可利用资源同样增多。

通过网络连接的主机之间的共享资源只有网络数据分组，修改数据分组自身使之在符合网络协议的前提下携带机密信息属于网络存储隐蔽信道；而如果利用数据分组传输过程中产生的时间特征，则会带来网络时间隐蔽信道。

在操作系统、数据库或者网络中发现一种共享资源作为隐蔽信道的传输介质，是隐蔽信道传输机制的核心，而好的传输介质的选择，能够提高信道的容量和隐匿性。隐蔽信道传输机制的设计充满了创新与想象，如何先于安全保护人员设计出一种难以察觉且能高效传输的信道，是隐蔽信道设计的重点。

3.2.2 编解码机制

提高隐蔽信道的传输准确率和隐匿性的另一种方式是改进信道的编解码机制[75,76]。Cabuk[65]和

Berk[67]的信道都采用 ASCII 编码。ASCII(american standard code for information interchange)是计算机系统中通用的单字节编码系统，主要用于显示现代英语和其他西欧语言。ASCII 码使用指定的 8bit 二进制数组合来表示 256 种可能的字符，因此 ASCII 的期待码长为 8bit。但文本信息统计结果表明，英文字母出现频率呈现规律性分布。对于隐蔽信道中传输的信息，字符频率也符合该分布。ASCII 码等编码没有利用字母的频率特征，编码期望长度较大。如果在编码过程中考虑频率特征，则会降低期望长度、减少隐蔽信道中传输的数据量。因此可以采用基于密码表的多元编码，如 Sellke[74]的 L bit to n packets 信道和基于 Huffman[12]编码的信道，利用敏感信息的编码冗余，降低传输信息量、缩短传输时间，从而提高信道容量。赫夫曼码[77]是一种典型的与频率相关的无损压缩码，为信源中出现最频繁的符号分配较短的码字，而为不经常出现的符号分配较长的码字，从而达到数据压缩的目的。

多元编码机制虽然增加了收发双方的编解码工作量，但整体上压缩了传输数据量、提高了信道容量。同时多元编码分散了共享资源属性特征出现频率，提高了隐匿性。如果配合纠错协议，能够大幅降低编码错误率。

4　云计算隐蔽信道实例与分类

UC Berkeley[31,36]大学的研究人员、HP[37]实验室的研究人员等都认为，由云计算环境中大量的共享资源导致的隐蔽信道，是云计算平台中最重要的安全威胁。虽然对云计算和隐蔽信道都有了一定的研究，但是如何准确分析云计算环境下的隐蔽信道仍然是当前迫切需要解决的问题。当前云环境下隐蔽信道研究都是基于经验的。例如，Thomas[17]、Okamura[18]等人对隐蔽信道的研究都是基于共享资源冲突的逆推结果，并不是基于对虚拟机系统的脆弱性分析，不是基于对云平台环境下隐蔽信道的系统化搜索。如何形式化可重复地搜索云平台隐蔽信道，并在云环境中为其构建场景、度量其威胁，在公开文献中还未见发表。

国内对云计算安全性的研究也日趋繁荣，2009 年清华大学的陈康和郑伟民[2]综述了当前云计算所采用的技术，具体分析了云计算的技术含义以及当前各企业主要采用的云计算实现方案。2010 年，中国科学院软件研究所的冯登国等人[39]分析了云计算对信息安全领域中技术、标准、监管等各方面带来的挑战；提出云计算安全参考框架及该框架下的主要研究内容；指出云计算的普及与应用是近年来信息安全领域的重大挑战与发展契机，将引发信息安全领域又一次重要的技术变革。

4.1　隐蔽信道实例

云计算平台以虚拟机为基础设施，提供了高度的隔离性，支持不同操作系统和应用程序同时运行。然而由于存在大量的共享资源，隐蔽信道问题则不可避免。下面以 Xen[5]平台中具体的隐蔽信道实例，说明云计算环境下隐蔽信道的具体实现以及攻击方式。

4.1.1　基于共享内存的隐蔽信道

Xen Hypervisor 是 Amazon EC2 平台的基础设施，位于操作系统和硬件之间，为上层的操作系统内核提供硬件基础设施支持[78~80]。为了完成虚拟机域间的通信与协作，Xen 提供了 2 类共享资源，即超级调用和事件通道[5]。由于 Xen 位于系统最高级 0 环，而 Guest OS 位于 1 环。对于 Guest OS 不能完成的特权操作，必须提供超级调用交由 Xen 代为完成。事件通道是 Xen 用于 Domain 和 Xen 之间、Domain 和 Domain 之间的异步事件通知机制。事件通道机制与超级调用机制一起完成 Xen 和 Domain 之间的控制和交互：使用超级调用产生从 Domain 到 Xen 的同步调用；使用异步事件机制完成从 Xen 到 Domain 的通知交递。这些共享资源构成了隐蔽信道的潜在源泉，例如，基于共享内存的时间隐蔽信道(SMCTC，sharing memory covert timing channel)就是其中一种典型的利用硬件资源共享的隐蔽信道[19]。

为了实现虚拟机 Domain 之间的共享内存，Xen 提供了基于超级调用和事件通道的授权表机制。每个 Domain 都拥有自己的授权表，DomA 创建一个环形数据结构并赋给其他虚拟域如 DomB 访问权限，以此构成共享内存。当 DomA 向共享内存中填充数据后，会通过异步通知机制通知 DomB 来访问数据，DomB 申请中断获取共享内存中的数据。在传输过程中，如果 DomA 控制共享内存填充时间，DomB 观察获取数据的时间，能够根据时间的不确定性特征，构成基于共享内存的时间隐蔽信道。

Joon 认为典型的隐蔽信道模型包括发送方和接收方的同步阶段、传输阶段和信息传输反馈阶段[81]。

在同步阶段发送方通知接收端同步信道相关信息，包括传输周期、编码方式等；在传输阶段阶段，发送方按照约定的编码方式将信息有序发送；在接收方反馈之后，发送方开启一个新的传输周期； 收发双方循环执行，直到所有的信息发送完毕。

在SMCTC信道中，环形共享内存结构作为传输中介，被DomA中的进程P_i和DomB中的进程P_j以消费者和生产者模型方式使用。SMCTC详细的通信协议实现过程如图4所示。

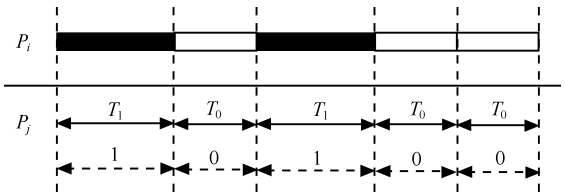

图4 基于共享内存虚拟机时间隐蔽信道场景

1) P_i将要传输的机密信息编码成二进制字符串，用T_0和T_1分别表示符号0和1且$T_0 < T_1$；

2) P_i建立环形共享数据结构并在授权表中添加授权项，将授权引用发送给P_j；

3) P_j将共享内存映射到自己的地址空间；

4) P_i将机密信息编码后的二进制字符串时间表示T_0和T_1插入到发送时间中，然后根据修改后的发送时间间隔序列发送待发送的原始数据；

5) 在每一个中断周期，P_j得到原始数据及其到达时间信息，并计算相应的时间间隔；

6) 当所有的机密信息发送完后，继续发送原始数据直到发送过程完成；

7) P_j得到所有的原始信息，将时间间隔序列逆向解析为二进制字符串，然后将其解码成P_i发送的机密信息，在通信周期完成之后，P_j解除共享内存映射；

8) P_i回收授权引用，隐蔽信息传输周期结束。

在图4中，通过SMCTC信道传输的机密信息被编码成10100，发送端根据编码后的信息结合T_0和T_1修改发送时间，接收方观察接收到的数据时间间隔。在Xen4.0.0平台上的实验结果表明，SMCTC信道容量能够达到174.98bit/s且错误率能够控制在2%的范围内，但实验操作同时表明SMCTC会受到同一物理平台上其他虚拟机的影响。在SMCTC的传输过程中，被传输的原始数据没有被修改，被修改的只有数据发送时间间隔，因此SMCTC信道能够跨越防火墙，入侵检测系统等，实现云计算环境下的信息泄漏。

4.1.2 基于Cache/CPU负载的隐蔽信道

2009年，Thomas等研究人员指出处在不同虚拟机之间的进程如果存在硬件资源共享就可能产生隐蔽信道[17]。可能被利用的共享资源包括网络接口、CPU分支预测表、指令Cache、内存总线、CPU调度器、CPU时间片、硬盘接口等。并在Amazon EC2平台上实现了基于内存总线和硬盘访问冲突的隐蔽信道，其信道容量分别为0.006bit/s和0.0005bit/s。同时，他们也提出一种基于物理Cache缓存的隐蔽信道传输方式。在该信道中，信道发送方用不执行操作和执行大量内存访问操作来表示信息0和1；接收方访问内存并观察访问延迟时间。如果延迟相对较大，说明接收端的内存访问需要先清除接收方Cache缓存，这意味着发送方执行了大量内存访问正在传输信息1；如果延迟相对较低，则表示发送方保持沉默，发送信息0。实验表明，虚拟机中隐蔽信道是确实存在的，并且云计算平台真正受到了隐蔽信道的威胁。

2010年，Okamura和Oyama深入研究了Thomas等人曾涉及到的基于CPU负载的隐蔽信道[18]。类似于基于Cache缓存的隐蔽信道，CPU负载信道也是观测进程执行操作的响应时间，用不同的时间表示不同的信息。他们的研究更加侧重于定量地分析信道的威胁，当云计算平台存在一个或多个物理CPU时，虚拟CPU对操作的分配对信道威胁产生的影响，以及在存在干扰的情况下，隐蔽信道的传输准确率。试验结果表明，在无干扰的情况下，CPU负载信道容量能够达到0.49bit/s，而在存在干扰时，信道准确率会减低10%。

SMCTC、Cache缓存信道和CPU负载信道，都是时间相关的信道。虽然在实验中都采用了ASCII编码方式，但是在时间识别精度允许的前提下，都可以扩展成多元编码机制，例如使用Huffman编码，从而设计更加健壮的容错协议来提高信道的容量、传输准确性和隐匿性[12]。

4.2 新的分类方式

隐蔽信道可以形式化地表述为$<V, PA_h, PV_1, P>$，变量V可以表示系统中共享资源的不同属性。当V表示存储属性时，隐蔽信道为存储隐蔽信道。例如，在事件标识型信道中，V表示收发双方能够修改和感知的事件标识值[13]。当V表示CPU时间或者其他与时间相关联的属性时，隐蔽信道为时间隐蔽信道。与存储隐蔽信道相比，时间隐蔽信道又称为无

记忆通道，不能长久地存储信息。发送者发送的信息接收者必须及时接收，否则要传递的信息就会消失，时效性较强[10]。

隐蔽信道本质上是信息的通信信道，因此可以分为噪音信道和无噪信道。如果 $<V, PA_h, PV_1, P>$ 中的变量 V 只能被进程 PA_h 修改，而且对于任意修改，进程 PV_1 都能实现概率为 1 的正确解码，则该信道称为无噪信道；如果变量 V 被进程 PA_h 修改的同时还可能被其他进程修改，导致 PV_1 解码出现错误，则该信道称为噪音信道。在隐蔽信道分析中，通常将信道抽象成无噪信道以度量信道容量最大值；但是在实际场景中，信道多为噪音信道，影响隐蔽信道的传输效率[10]。

现有的研究对隐蔽信道的分类基本上都是出于对工程实践的考虑。例如，存储隐蔽信道和时间隐蔽信道并没有本质的区别[50]，只是变量 V 代表的属性不同。但是在实际标识、检测的过程中，时间隐蔽信道难度更高，因此安全标准只有在更高级的安全环境中才要求进行时间隐蔽信道分析[44~47]。根据是否存在噪音对隐蔽信道进行分类，方便计算信道的容量与威胁。然而在云计算环境中，隐蔽信道问题更加复杂(如图 5 所示)，现有的分类方式并没有体现出隐蔽信道在云计算环境中的特点，所以需要新的分类方式以更准确地刻画隐蔽信道的外延和内涵。

4.2.1　危害影响范围分类方式

虚拟化技术允许在同一硬件平台上创建多个独立虚拟域，分别向用户提供服务。云计算以虚拟化技术为基础，管理同一硬件平台和跨硬件平台的虚拟机逻辑资源。按照隐蔽信道危害的影响范围(以 Xen 虚拟化平台为例，如图 5 所示)，可以将隐蔽信道分为域内隐蔽信道(CC1)、跨平台隐蔽信道(CC2)和域间隐蔽信道(CC3)。

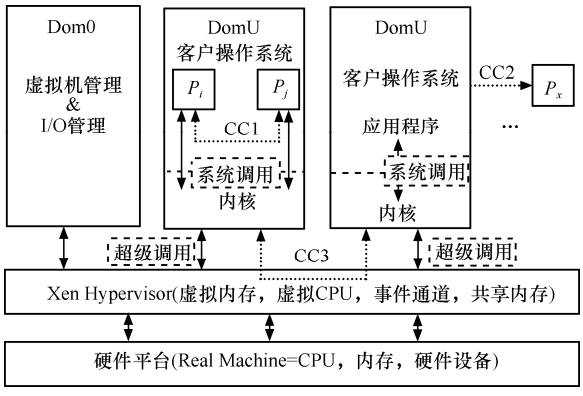

图 5　Xen 和典型的隐蔽信道场景

1) 域内隐蔽信道 CC1，进程级泄漏方式。恶意进程 P_i 和 P_j 处在同一虚拟域(DomU)中，由于虚拟机提供的强隔离性机制，隐蔽信道影响的范围局限在该虚拟域内。DomU 中运行独立的操作系统，P_i 和 P_j 是处于不同安全级的操作进程，机密信息从高等级进程 P_i 泄漏到低等级进程 P_j，从而实现隐蔽信道通信。CC1 类型的隐蔽信道是操作系统中的进程级机密信息泄漏方式，对其的标识、度量、消除、限制、审计和检测等方法可参考操作系统隐蔽信道的分析方法。

2) 跨平台隐蔽信道 CC2，网络级隐蔽信道。恶意进程 P_k 在虚拟机平台 DomU 中，P_x 是其他硬件平台的虚拟机或者独立操作系统中的进程。进程 P_k 和 P_x 只能通过网络连接通信，因此 CC2 信道可抽象为网络隐蔽信道。对 CC2 型信道的研究可以参考传统的网络隐蔽信道研究方法。

3) 域间隐蔽信道 CC3，系统级泄漏方式。收发双方恶意进程分处同一硬件平台上不同的虚拟域中，机密信息经过操作系统级的传输，泄漏给恶意用户。CC3 类型的隐蔽信道是云计算环境中特有的隐蔽信道类型，是由硬件资源共享导致的信道，如基于共享内存、Cache 和 CPU 负载的信道。CC3 信道对于云计算客户的数据安全至关重要，如果具有业务竞争关系的客户处在同一物理平台上，CC3 类型的信息泄漏将带来沉重的经济代价。

4.2.2　层次分类方式

按照提供服务的层次，云计算平台自顶向下分为软件应用层、平台层以及基础设施层。相应地，隐蔽信道可以按照层次划分为应用层隐蔽信道(CC1，CC2)和基础设施层隐蔽信道(CC3)。

1) 应用层隐蔽信道(CC1，CC2)，信道的收发双方利用操作系统、应用软件、网络程序固有的共享资源属性构建隐蔽信道传输机密信息。对应用层隐蔽信道的研究可参考现有的针对操作系统、数据库以及网络隐蔽信道的分析方法。

2) 基础设施层隐蔽信道(CC3)。在云计算平台中，虚拟机系统控制硬件资源，可视为广义上的操作系统，为上层的抽象进程　　虚拟机提供服务。CC3 类型的隐蔽信道利用虚拟机和硬件中的共享资源属性创建隐蔽信道传输机密信息。

层次分类方式将 CC1、CC2 划分为同一个类别与 CC3 进行区分，有助于隐蔽信道分析工程实践。到目前为止，大部分的隐蔽信道研究工作都针对

CC1、CC2类型的隐蔽信道，有相对成熟的分析方法和解决方案；CC3型隐蔽信道是云计算环境中的新问题。分析云计算平台隐蔽信道时，可以对CC3做增量分析，再结合CC1、CC2的分析结果，从而进一步提高隐蔽信道分析的效率和工程进度。

5 云计算隐蔽信道关键问题

云计算凭借其对服务和计算资源的高效管理和弹性分配，充分体现了“网络就是计算机”的思想[39]。虽然虚拟化技术为同一硬件平台上不同客户之间提供数据隔离保障，隐蔽信道机制仍然能够破坏云计算平台的隔离性，泄漏客户的机密信息[17]。在云计算这种复杂的新型网络环境中，如何对隐蔽信道进行全面准确地分析，主要面临以下4个问题。

1）缺乏云计算平台中隐蔽信道的形式化定义。在将近40年的隐蔽信道研究中，研究人员对隐蔽信道的研究可谓是仁者见仁，智者见智，其定义也分别侧重于信息流、格策略模型和访问控制等[51]。而在云计算环境中，如何对隐蔽信道进行合理的定义，并用这种定义指导分析实践是首要解决的问题。

2）缺乏针对云计算平台的系统化可操作的隐蔽信道标识方法。传统的隐蔽信道标识方法主要集中于对操作系统的分析，分析的切入点包括用户手册、形式化顶层规范、详细顶层规范以及系统源代码。标识的方法包括语义/语法信息流方法、共享资源矩阵法和无干扰方法等，然而这些方法在实践中都存在一定的局限性[10]。云计算作为计算资源的管理模式，广义上可以看做一种新型的操作系统。针对这种新型的操作系统，尚缺乏系统化的标识方法。

3）缺乏针对云计算平台中隐蔽信道威胁的度量方法。隐蔽信道是系统的潜在威胁，如何准确评估其威胁程度，是深入了解系统脆弱性、并采取一定的补救措施的前提。传统的度量方法，主要考虑信道容量指标[10]。而在云计算环境中，多虚拟机之间的相互干扰[18]、信道编码机制的选取[12,74]，都会影响信道容量，也会影响信息传输的准确率。如何综合评估云计算环境下的隐蔽信道威胁，在真实的信道场景中，仍然缺乏可实践的度量方法。

4）缺乏针对云计算平台中隐蔽信道分析的安全标准。1985年美国国防部颁布的TCSEC[44]可信计算基安全标准、1999年国际标准化组织颁布的CC[45]标准以及GB17859-2001[46]、GB/T20272-2006[47]等我国的安全标准都要求安全信息系统必须进行隐蔽信道分析。但这些标准主要针对传统操作系统、安全数据库和网络系统，到目前为止还没有一部针对云计算平台安全的标准出台。随着云服务商和客户对安全和隐私保护要求的日益强烈，标准的拟定推行势在必行。如何将隐蔽信道分析在云计算平台上标准化，是当前学术界和工业界研究的必然方向。

5.1 云计算隐蔽信道定义

在操作系统隐蔽信道研究中，隐蔽信道被形式化表述为$<V,PA_h,PV_l,P>$[10,13]。该定义强调了操作系统中具有不同安全级的恶意进程PA_h和PV_l合谋借助共享资源V进行违反安全策略P的通信。定义中的4个要素是操作系统中任何隐蔽信道必须具备的基本条件，是隐蔽信道分析工程实践的指导标准。例如，信道标识过程即在系统中搜索满足定义要求的进程PA_h和PV_l；信道度量首先为PA_h和PV_l创建符合系统情况的场景，然后度量其最大威胁；信道消除和限制强调消除V的存在，或者限制PA_h和PV_l的执行。

云计算环境涉及到虚拟机监控器、虚拟机以及虚拟机中相对独立的操作系统，其安全特征主要表现为虚拟机的隔离性，如何以形式化的方式对隐蔽信道进行标识变得愈加困难。给出一个相对全面的隐蔽信道定义，能够概括云计算环境下隐蔽信道(如图5中CC1、CC2和CC3)的特征、内延和外涵，从而指导隐蔽信道分析实践方案的设计，是云计算隐蔽信道研究首要的内容。

Denning提出从信息流的角度分析隐蔽性，是其他分析方法的思路源泉。研究表明，隐蔽信道本质上是非法信息流，因此定义和研究都可以以信息流为基础[82~85]。参考Denning信息流的定义方法，本文将云计算虚拟机(以Xen为例)信息流模型定义为

$$XFM=<N,P,VM,Q,\rightarrow>$$

其中，$N=\{a,b,\cdots\}$是包含了共享逻辑存储资源和时间资源的集合；用$s_a=\{a_1,a_2,\cdots\}$表示共享资源对象a的状态值集合。在操作系统隐蔽信道中，逻辑资源集合N中的元素可以是文件、事件状态等；集合s_a表示每个元素的值域。在云计算平台的基于CPU负载和Cache缓存的隐蔽信道中，逻辑资源集合N中的元素分别表示vCPU、Cache等共享资源。

在$XFM=<N,P,VM,Q,\rightarrow>$中，P是进程的集合，表示为$P=\{P_1,P_2,\cdots\},Dom(P_i),Dom(P_j)\ VM$，其中$VM=\{VM_1,VM_2,\cdots\}$表示进程所在的虚拟机系

统；P_i 和 P_j 处在同一个虚拟机，表示为 $Dom(P_i)$，$Dom(P_j)$ VM_k，i、j、k 分别表示进程序号和虚拟机序号。当 P_i 和 P_j 在同一个虚拟机内部时，产生的隐蔽信道属于 CC1 类型，处在同一硬件平台不同虚拟机之间时属于 CC3 类型，不在同一硬件平台时属于 CC2 类型。

二元操作符 Q 表示进程 P_i 能够修改/访问逻辑对象 a，且修改后其值仍满足其值域，即 P_iQa S_a。二元关系操作符 表示信息流自左向右流动，P_iQa P_jQb 表示进程 P_i 通过操作逻辑对象 a 发送信息，P_j 通过操作逻辑对象 b 接收信息。

通过对虚拟机系统信息流模型的描述，可定义虚拟机的隔离性，即虚拟机系统的无干扰模型为：

$$\forall a \in N, P_iQa = P_jQ(P_iQa)$$

表示对于任意的逻辑对象 a，共享资源进程 P_j 无法通过 a 推测进程 P_i 的任何操作，即没有信息流从 P_i 到 P_j，表示为 $P_i \nrightarrow P_j$。

然而，在云计算环境中，虚拟机之间是相互隔离无干扰的。隐蔽信道机制违反了系统的无干扰模型，即使部署了安全策略，仍然能够实现机密信息泄漏。因此，云计算隐蔽信道定义为：

$$\exists a \in N, P_iQa \overset{Sec}{} P_jQb$$

其中，$\overset{Sec}{}$ 表示云计算虚拟机系统中部署了安全策略，在该策略下 P_i 和 P_j 无干扰($P_i \nrightarrow P_j$)，操作对象 a 和 b 逻辑独立，但是在实际系统中，a 和 b 可以是相同的物理资源，例如 CPU 资源。P_i 和 P_j 通过操作逻辑对象 a 和 b 实现违反无干扰模型的通信，而 P_i 和 P_j 之间的这种类型的通信信道就称为隐蔽信道。

云计算隐蔽信道定义体现了隐蔽信道的必要元素，包括收发进程、操作对象、信息流模型以及无干扰模型，完整表达了隐蔽信道在云计算环境下的特征。对云计算平台进行信息流分析，对云计算基础设施、平台及应用层进行无干扰模型分析，任何满足该定义的通信信道都是潜在的隐蔽信道。

5.2　云计算隐蔽信道标识方法

云计算隐蔽信道定义抽象概括了虚拟机系统中的 3 种隐蔽信道类型，包括 CC1、CC2 和 CC3。层次分类方式按照隐蔽信道所处位置将 CC1、CC2 与 CC3 区分，有助于隐蔽信道标识的工程实践。对于应用层隐蔽信道(CC1，CC2)类型的隐蔽信道，有相对成熟的标识方法；CC3 类型的标识可以参考现有方法，做增量分析，从而提高分析效率和工程进度。

系统源代码包含了所有安全相关信息，穷举式地分析系统源代码虽然全面，但是规模庞大、实现复杂。Denning[52]、Tsai[51]、Kemmerer[48,50]等人的方法都存在工作量大或者状态爆炸等问题。如何避免和解决这些问题，是隐蔽信道标识问题研究的重点。

在研究中，文献[13]提出了基于源代码的有向信息流图标识方法，具体步骤如下：

1）按照高内聚、低耦合的规则将规模庞大的系统划分为相对独立的多个子系统；

2）采用 LLVM[55]编译技术，将分析对象编译成等价的更具结构性的中间代码；

3）针对中间代码设计查找算法，分析模块中共享资源和操作进程；

4）结合信息流分析技术，为查找到的共享资源和操作进程创建有向信息流图；

5）设计有向图搜索算法、剪枝算法，查找满足隐蔽信道定义的共享资源，即为潜在的隐蔽信道。

将完整的系统划分成相对独立的子系统能够降低分析规模，不同的子系统可分配给不同的工作组并行分析，在一定程度上降低了分析复杂度；对独立子系统分析之后，要合并子系统的分析结果，如果存在跨子系统的潜在共享资源，要做进一步的归纳分析。

该方法紧密结合隐蔽信道定义，其有效性在操作系统隐蔽信道标识中已经得到证实；对于符合云计算隐蔽信道定义的信道会同样适用。该方法的技术难点在于使用 LLVM 进行等价编译，并对查找到的共享资源和进程构建有向信息流图。LLVM 编译后的中间代码保持了源代码的所有特征，以模块、函数、基本块、指令、操作数组织代码，更具结构化。通过设计共享资源查找算法，能够快速、准确全面地从中间代码中查找隐蔽信道的必要元素。查找结果以共享资源为中心，操作进程为分支，构建有向信息流图。如果图中有 2 个及以上的修改/访问分支能够被应用层调用，就可能构成隐蔽信道，此图则作为分析结果输出。当系统源代码都经过搜索之后，输出结果集合即为云计算平台潜在隐蔽信道集合。

隐蔽信道定义指导信道标识方法的设计。云计算环境下的隐蔽信道标识方法通过对虚拟机平台的系统化搜索，能发现所有潜在的隐蔽信道，并且

该方法需要较少的人工干预，能实现最大程度的自动化。

5.3 云计算隐蔽信道场景构建及度量方法

为了准确地度量隐蔽信道的威胁，必须在具体的环境中为隐蔽信道构建场景。现有文献对如何构建隐蔽信道场景的研究较少，而是将场景构建合并到隐蔽信道设计的研究之中。构建隐蔽信道场景能够验证潜在隐蔽信道是否能在云计算环境下真实实现，场景构建过程中涉及到的因素对信道威胁会产生何种影响，应该作为信道威胁评估的前提和重点。

操作系统中存在一种事件标识型信道，该信道的收发双方通过改变和观察特定事件的状态，合谋传递机密信息。事件标识型信道场景可以建模为非确定型有限状态机，如图6所示[13]。信道的初始状态为q_0，信道的收发双方约定固定的时间周期。在一个周期内，如果发送方改变事件状态，则表示传输信息1，有限状态机发生相应的变化，即q_0 q_1。在q_1状态下，如果发送方持续发送信号1，则事件状态值始终为q_1。在q_0和q_1状态下，如果发送方发送信号0，事件状态转向q_2，即q_0 q_2，q_1 q_2。当传输过程结束时，事件状态为结束状态q_2或q_3。接收方观察事件的状态值，解码机密信息。收发双方的合谋协作实现机密信息的泄漏。

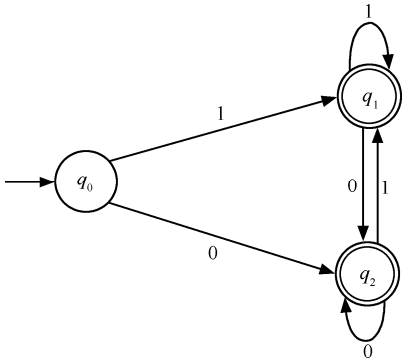

图6 事件标识型隐蔽信道场景模型

隐蔽信道利用共享资源的存储属性或者时间属性构建场景，根据其属性特征及信号分布，可以建模成有限状态自动机作为信道度量的基础。在云计算环境下构建隐蔽信道场景之后，如何综合准确地评估其威胁对系统入侵者和系统安全管理员来说都是至关重要的，度量结果可用来指导信道限制措施的实施。对于入侵者来说，期望尽可能地提高信道的性能，能够在较短的时间内传输尽可能多的机密信息，同时入侵者期望这些信息在传输过程中具有较高的准确率。而对于安全管理员来说，则要做到知己知彼，能够评估该信道可能被利用的最大程度，即评估信道的威胁最大值；发现影响信道威胁的因素，并利用这些影响因素施加消除或者干扰措施从而降低信道对系统的威胁。因此，信道度量是整个信道分析过程中的关键环节。虽然安全标准中只强调了信道的容量，但是对于攻防双方来说，信息的敏感度和保真度都是度量隐蔽信道威胁的重要指标，度量信道威胁的指标主要包括[10]：

1）信道容量。信道能够取得的最大信息传输速率，容量的概念最早来源于信息理论，并被TCSEC采纳作为信道威胁评价的指标，因此在信道度量领域应用最为广泛，计算容量的方法中最为著名的是Millen[86]提出的形式化方法和Tsai[87]等人提出的非形式化方法，这2种方法都得到了TCSEC标准的认可。

2）事务安全级别差。Ahmed[88]利用冲突事务的安全级别差作为信道威胁的度量，并命名为隐蔽信道因素，信道两端的安全级别差别越大，其间的隐蔽信息传输对系统的威胁就越大。

3）短消息指标。信道容量适合描述信道传输长文件的能力，但是并不适合度量信道的短消息传输能力。针对信道容量指标的缺陷，Moskowitz[89]提出了短消息指标的概念，利用消息长度参数n、消息传输时间t和消息保真度r共同描述信道的短消息传输能力。文献[14]对Moskowitz的短消息指标加以改进，综合考虑了消息的传输时间、保真度以及敏感度因素，设计了一个统一的价值函数表示系统对信道短消息传输能力评估，其价值函数计算方法为：$V_i(r,t)=(r\%)^W U_i(t)$，其中短消息传输价值V_i是入侵者利用隐蔽信道在时间t内，以保真度r完成对数据项d_i的传输所获得的价值，$U_i(t)$为数据项d_i的准确信息的价值随时间变化的函数。该指标与容量指标相结合，能够综合度量数据库系统中的隐蔽信道威胁。

在云计算环境中，影响隐蔽信道性能的因素很多，包括硬件资源共享冲突、网络环境、硬件性能等。现有的研究表明，在虚拟机环境下，将虚拟机任务分配到多核CPU不同的物理核心上，会降低信道的干扰；从另一个角度讲，如果CPU负载信道的收发双方被分配到不同的物理核心上，则传输信息的错误率会大幅提高。如何准确评估硬件资源分配对信道的影响是一个复杂的问题。另一个影响

隐蔽信道性能的因素是系统的编解码机制，用敏感信息的编码冗余，降低传输信息量，缩短传输时间，从而提高信道容量。

到目前为止，云计算环境尚缺乏一种能够综合考虑各种影响因素，量化信道容量、信道保真度、编码机制、纠错协议、网络及硬件等特性的方法。如何设计这种综合指标是今后云计算隐蔽信道度量技术研究的重点。

5.4　云计算隐蔽信道安全标准

安全问题始终是信息系统面临的最大问题，随着云计算的发展，安全问题的重要性更加突出。对安全标准的需求更加迫切，一部完整的安全标准不仅能够指导云计算平台的应用服务开发，也能够评估现有云计算平台是否达到业界的安全标准需求。

国内外的安全标准对安全信息系统隐蔽信道分析都做了明确的要求，如 TCSEC[44]、CC[45]、GB17859-2001[46]、GB/T20272-2006[47]等。针对安全操作系统，GB/T20272-2006[47]标准给出了隐蔽信道分析的详细且可操作的要求，其第 4 级标准主要包括：通过非形式化的搜索，标识出可识别的存储隐蔽信道，估算其带宽，记录存储隐蔽信道分析期间所作的全部假设，用封锁、限制带宽或审计等，对所标识的存储隐蔽信道进行处理。第 5 级标准要求更加严格，添加了形式化搜索和时间隐蔽信道的标识和处置。但是目前还没有一部针对云计算安全分析的标准。

2009 年，IBM、Cisco、SAP、EMC、RedHat、AMD、AT&T、VMware 等近百家 IT 企业发布了“开放式云宣言”[90]，总结了云计算的特点和挑战，提出建立开放的云基础设施是未来云计算领域的发展趋势。该宣言期望为业界制定开放的、统一的云计算标准，促进云计算公共事业的发展，最终造福于用户[1]。

云计算标准还处在酝酿阶段，随着云计算产业日趋庞大，其安全性日趋重要。如何以安全标准的形式保障云服务商和云客户的权益，保障客户数据不被泄露，成为日益紧迫的任务。隐蔽信道是云计算平台的重要威胁，因此期望安全标准从隐蔽信道分析的角度深入研究，为安全标准的制定提供有益的参考。

6　研究前景

6.1　学术价值

隐蔽信道与系统中信息流的安全模型或者完整性模型都密切相关。在实现强制安全策略模型的系统中，隐蔽信道允许信息流从特定高安全级到低安全级或者不可比安全级的非法泄密；在实现强制完整性模型的系统中，隐蔽信道允许信息流从特定高完整级到低完整级或者不可比完整级的非法泄密。如何限制云计算环境下隐蔽信道的威胁，确保云客户信息不被泄露，一直是学术界研究的重点[91]。

研究人员从云安全体系框架出发，设计了 Terra[92]、sHype[6,7]、Lares[38]、HyperSentry[9]、HyperSafe[43]等安全平台，保障虚拟机系统数据安全。

Terra[92]基于 TCG (trusted computing group)的可信计算硬件平台，继承了虚拟机平台的隔离、扩展、高效、兼容和安全性的特点，为同一个硬件平台上的独立虚拟域提供了完整性保护。Terra 平台引入 TVMM (trusted virtual machine monitor)的概念，为上层的独立虚拟域服务提供原始的硬件接口、隔离性、完整性保护和认证功能。开发人员根据应用程序的需求定制专用操作系统，满足特定安全需求。

sHype[6,7]实现了同一硬件平台上虚拟机之间的信息流控制模型，为虚拟机数据提供了安全策略保护。sHype 扩展了 Hypervisor 的隔离性，提供了启动和运行时保障，添加了安全机制，满足了共享资源在虚拟机之间的受控机制。虽然 sHype 提供了强隔离的安全机制，允许可信和不可信的操作系统在同一硬件平台上同时运行。但 sHype 仍不能彻底消除隐蔽信道，例如，基于共享内存的信道、CPU 负载信道、Cache 缓存等信道。

Lares[38]提出了一种针对虚拟机系统的主动监控架构。Lares 形式化了虚拟机平台面临的潜在威胁，包括旁路攻击、内容篡改、执行和返回过程被修改等攻击方式。为了能够使虚拟化平台在性能影响可接受的范围内保护被监控组件，Lares 设计了灵活的钩子点插入机制，创建了独立的安全虚拟域，对系统的行为模式进行主动监控，从而保证数据的完整性。

学术界对云安全的研究侧重于体系结构的构建，但是隐蔽信道问题一直都是研究的重点。除了 Terra[92]、sHype[6,7]、Lares[38]之外，还有 HyperSentry[9]、HyperSafe[43]、HIMA[93]、SecVisor[94]、REDAS[95]、Patagonix[96]、TCvisor[97]等都是面向虚拟机数据完整性保护的安全机制。从形式化角度证明系统的安全性，从而依据形式化证明结果实现系统是云安全的重要研究方向，例如 seL4[98]，Xenon[99~101]等。如

何应用虚拟机构建安全防护系统也是当前学术界的热点问题，例如，使用虚拟机平台构建入侵检测系统[102]等。云计算安全开创了学术界研究的新领域，其热度在未来几年将会一直持续。

6.2 工业价值

云计算掀起了当今 IT 界的又一次研究热潮，产业界对云计算能够带来的实际效益更加注重。Amazon、IBM、VMware、Google、Ctrix、Salesforce.com 以及其他所有 IT 公司都在谋划从云计算平台中盈利。然而对于云用户来说，最关注的依然是数据的安全，即如何保障数据在云平台中不被泄露。因此，隐蔽信道分析对云计算产业安全至关重要，如何隔离用户数据，保证数据的机密性、完整性以及可用性将会是今后工业界的研究重点。

7 结束语

研究表明，中国云计算产业规模 2012 年将突破 600 亿元。未来三年，国内云计算应用将以政府、电信、教育、医疗、金融、石油化工和电力等行业为重点，云计算在中国市场逐步被越来越多的企业和机构采用，将成为一项不可或缺的基础设施。云计算产业将会按照基础设施服务、平台服务和应用服务对云计算厂商进一步细分。如何在云计算产业发展浪潮中，把握关键技术、解决核心问题是研究人员的工作重点。对于使用共同云服务的具有竞争关系的云客户来说，机密信息的泄漏将会造成无法挽回的经济损失。而隐蔽信道就是这种危害性极强的信息泄漏途径。隐蔽通道研究是信息安全的一个重要难题。

本文立足于云计算安全战略需求，综述了云计算平台的发展现状，详细介绍了云计算平台架构、核心虚拟化技术及安全问题。对于操作系统、数据库和网络中的隐蔽信道基本问题深入浅出地总结了研究成果和现状。同时指出，在云计算平台下，隐蔽信道问题依然存在，并给出具体的实例。本文提出了一种虚拟机系统中基于共享内存的时间隐蔽信道，并从全新的角度对隐蔽信道进行分类，且该分类方式更适用于隐蔽信道分析操作。云计算隐蔽信道是云计算环境下至关重要的科学前沿问题，体现学科的交叉和综合。目前国内外的研究都处于起步阶段，尚有很大的研究余地和潜力。本文全面总结了云计算隐蔽信道研究中面临的 4 大问题，并给出云计算隐蔽信道的形式化定义，该定义从信息流角度出发，能够指导隐蔽信道标识、度量的工程实践。最后从学术和工业角度指出云计算隐蔽信道的研究价值。

参考文献：

[1] 虚拟化与云计算小组. 虚拟化与云计算[M]. 北京：电子工业出版社，2009.
Group of Virtualization and Cloud Computing. Virtualization and Cloud Computing[M]. Beijing: Publishing House of Electronics Industry, 2009.

[2] 陈康，郑纬民. 云计算:系统实例与研究现状[J]. 软件学报，2009，20(5):1337-1348.
CHAN K, ZHENG W M. Cloud computing: system instances and current research[J]. Journal of Software, 2009, 20(5): 1337-1348.

[3] NANCE K, BISHOP M, HAY B. Virtual machine introspection: observation or interference?[J]. IEEE Security and Privacy, 2008, 6(5): 32-37.

[4] MELL P, GRANCE T. The Nist Definition of Cloud Computing[S]. National Institute of Standards and Technology, 2009,

[5] BARHAM P, DRAGOVIC B, FRASER K, et al. Xen and the art of virtualization[A]. Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles[C]. Bolton Landing, NY, USA. 2003: 164-177.

[6] SAILER R, JAEGER T, VALDEZ E, et al. Building a MAC-based security architecture for the Xen open-source hypervisor[A]. Proceedings of the Computer Security Applications Conference[C]. Washington, DC, USA. 2005.276-285.

[7] SAILER R, VALDEZ E, JAEGER T, et al. sHype: Secure Hypervisor Approach to Trusted Virtualized Systems[R]. IBM Research Report RC23511, 2005, 1-12.

[8] PAYNE B D, SAILER R, CACERES R, et al. A layered approach to simplified access control in virtualized systems[J]. SIGOPS Oper Syst Rev, 2007, 41(4): 12-19.

[9] AZAB A M, NING P, WANG Z, et al. HyperSentry: enabling stealthy in-context measurement of hypervisor integrity[A]. Proceedings of the 17th ACM conference on Computer and communications security[C]. Chicago, Illinois, USA. 2010: 38-49.

[10] 王永吉，吴敬征，曾海涛等. 隐蔽信道研究[J]. 软件学报，2010，21(9):2262-2288.
WANG Y J, WU J Z, ZENG H T, et al. Covert channel research[J]. Journal of Software, 2010, 21(9): 2262-2288.

[11] BUTLER W L. A note on the confinement problem[J]. Commun ACM, 1973, 16(10): 613-615.

[12] WU J Z, WANG Y J, DING L P, et al. Improving performance of network covert timing channel through huffman coding[A]. The 2010 FTRA International Symposium on Advances in Cryptography, Security and Applications for Future Computing (ACSA 2010)[C]. Gwangju, Korea. Dec 9-11, 2010. 512-521.

[13] WU J Z, WANG Y J, DING L P, et al. Constructing scenario of

event-flag covert channel in secure operating system [A]. 2nd International Conference on Information and Multimedia Technology (ICIMT 2010)[C]. Hongkong. Dec 28-30, 2010: 371-375.

[14] 曾海涛, 王永吉, 祖伟等. 短消息指标新定义及在事务信道限制中的应用[J]. 软件学报, 2009, 20(4): 985-996.
ZENG H T, WANG Y J, ZU W, et al. New definition of small message criterion and its application in transaction covert channel mitigating[J]. Journal of Software, 2009, 20(4): 985-996.

[15] 曾海涛, 王永吉, 阮利等. 使用容量指标的安全实时数据库信道限制方法[J]. 通信学报, 2008, 29(8): 46-56.
ZENG H T, WANG Y J, RUAN L, et al. Covert channel mitigation method for secure real-time database using capacity metric[J]. Journal on Communications, 2008, 29(8): 46-56.

[16] ZANDER S, ARMITAGE G, BRANCH P. A survey of covert channels and countermeasures in computer network protocols[J]. Communications Surveys & Tutorials, IEEE, 2007, 9(3): 44-57.

[17] RISTENPART T, TROMER E, SHACHAM H, et al. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds[A]. Proceedings of the 16th ACM conference on Computer and communications security[C]. Chicago, Illinois, USA, 2009. 199-212.

[18] OKAMURA K, OYAMA Y. Load-based covert channels between Xen virtual machines[A]. Proceedings of the 2010 ACM Symposium on Applied Computing. Sierre[C]. Switzerland, 2010. 173-180.

[19] WU J Z, DING L P, WANG Y J, et al. Identification and evaluation of sharing memory covert timing channel in xen virtual machines[A]. Proceedings of the Cloud Computing (CLOUD), 2011 IEEE International Conference on[C]. Washington, DC, USA, 2011. 283-291.

[20] AVIRAM A, HU S, FORD B, et al. Determinating timing channels in compute clouds[A]. Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop[C]. Chicago, Illinois, USA, 2010. 103-108.

[21] LI S, EPHREMIDES A. Covert channels in ad-hoc wireless networks[J]. Ad Hoc Netw, 2010, 8(2): 135-147.

[22] 卿斯汉, 沈昌祥. 高等级安全操作系统的设计[J]. 中国科学(E 辑: 信息科学), 2007, 37(2): 238-253.
QING S H, SHEN C X. Design of secure operating systems with high security levels[J]. Science in China Series E: Information Sciences, 2007, 37(2): 238-253.

[23] 卿斯汉. 高安全等级安全操作系统的隐蔽通道分析[J]. 软件学报, 2004, 15(12): 1837-1849.
QING S H. Covert channel analysis in secure operating systems with high security levels[J]. Journal of Software, 2004, 15(12): 1837-1849.

[24] QING S H, SHEN C X. Design of secure operating systems with high security levels[J]. Science in China Series F: Information Sciences, 2007, 50(3): 399-418.

[25] WU J Z, DING L P, WANG Y J, et al. A Practical covert channel identification approach in source code based on directed information flow graph[A]. Proceedings of the Secure Software Integration and Reliability Improvement (SSIRI), 2011 Fifth International Conference on[C]. Jeju Island, Korea. 2011. 98-107.

[26] BISHOP S, OKHRAVI H, RAHIM I S, et al. Covert channel resistant information leakage protection using a multi-agent architecture[J]. Information Security, IET, 2010, 4(4): 233-247.

[27] WANG Y, CHEN P, GE Y, et al. Traffic controller: a practical approach to block network covert timing channel[A]. Proceedings of the Availability, Reliability and Security, 2009 ARES '09 International Conference on[C]. 2009. 349-354.

[28] SENGUPTA S, ANAND S, HONG K, et al. On adversarial games in dynamic spectrum access networking based covert timing channels?[J]. SIGMOBILE Mob Comput Commun Rev, 2009, 13(2): 96-107.

[29] KELLY K. Out of Control[M]. Beijing: New Start Press, 2010.

[30] BOSS G, MALLADI P, QUAN D, et al. Cloud computing[R]. IBM Whitepaper. 2007.

[31] ARMBRUST M, FOX A, GRIFFITH R, et al. UCB/EECS-2009-28[R]. EECS Department, University of California, Berkeley, 2009.

[32] ARMBRUST M, FOX A, GRIFFITH R, et al. A view of cloud computing[J]. Commun ACM, 2010, 53(4): 50-58.

[33] Amazon elastic compute cloud (Amazon EC2)[EB/OL]. http://aws.amazon.com/ec2/. 2009.

[34] Google app engine[EB/OL]. http://code.google.com/appengine/. 2009.

[35] Business CRM Solutions[EB/OL]. http://www.salesforce.com/crm/.

[36] CHEN Y, PAXSON V, KATZ R H. UCB/EECS-2010-5[R]: EECS Department, University of California, Berkeley, 2010.

[37] VAQUERO L, RODERO-MERINO L, MORN D. Locking the sky: a survey on IaaS cloud security[J]. Computing, 2011, 91(1): 93-118.

[38] PAYNE B D, CARBONE M, SHARIF M, et al. Lares: an architecture for secure active monitoring using virtualization[A]. 2008 IEEE Symposium on Security and Privacy[C]. Oakland, CA, 2008. 233-247.

[39] 冯登国, 张敏, 张妍等. 云计算安全研究[J]. 软件学报, 2011, 22(1): 71-83.
FENG D G, ZHANG M, ZHANG Y, et al. Study on cloud computing security[J]. Journal of Software, 2011, 22(1): 71-83.

[40] 诸葛建伟, 韩心慧, 周勇林等. 僵尸网络研究[J]. 软件学报, 2008, 19(3): 702-715.
ZHUGE J W, HAN X H, ZHOU Y L, et al. Research and development of botnets[J]. Journal of Software, 2008, 19(3): 702-715.

[41] GRIER C, SHUO T, KING S T. Secure Web browsing with the OP Web browser[A]; 2008 IEEE Symposium on Security and Privacy[C]. Oakland, CA, 2008. 402-416.

[42] CHEN S, WANG R, WANG X, et al. Side-channel leaks in web applications: a reality today, a challenge tomorrow[A]. 2010 IEEE Symposium on Security and Privacy[C]. 2010. 191-206.

[43] WANG Z, JIANG X. HyperSafe: a lightweight approach to provide lifetime hypervisor control-flow integrity[A]. 2010 IEEE Symposium on Security and Privacy[C]. 2010. 380-395.

[44] CENTER N C S. Trusted Computer System Evaluation Criteria[S]. 1985.

[45] ISO/IEC. Common Criteria for Information Technology Security Evaluation[S]. ISO Online Catalogue, 2005.

[46] GB 17859-1999. 计算机信息系统安全保护等级划分准则[S]. 2001.
GB 17859-1999. Classified Criteria for Security Protection of Computer Information System [S]. 2001.

[47] GB/T 20272-2006. 信息安全技术操作系统安全技术要求[S]. 2006.
GB/T 20272-2006. Information Security Technology - Security Techniques Requirement for Operating System [S]. 2006.

[48] KEMMERER R A. A practical approach to identifying storage and timing channels: twenty years later[A]. 18th Annual Computer Security Applications Conference[C]. LAS VEGAS, NV, 2002. 109-118.

[49] KEMMERER R A, PORRAS P A. Covert flow trees: a visual approach to analyzing covert storage channels[J]. IEEE Transactions on Software Engineering. 1991, 17(11): 1166-1185.

[50] KEMMERER R A. Shared resource matrix methodology: an approach to identifying storage and timing channels[J]. ACM Trans Comput Syst, 1983, 1(3): 256-277.

[51] TSAI C-R, GLIGOR V D, CHANDERSEKARAN C S. A Formal Method for the Identification of Covert Storage Channels in Source Code[A]. 1987 IEEE Symposium on Security and Privacy[C]. Oakland, CA, 1987. 74-86.

[52] DENNING D E. A lattice model of secure information flow [J]. Commun ACM, 1976, 19(5): 236-243.

[53] 卿斯汉, 朱继锋. 安胜安全操作系统的隐蔽通道分析[J]. 软件学报, 2004, 15(09): 1385-92.
QING S H, ZHU J F. Covet channel analysis on ANSHENG secure operating system [J]. Journal of Software, 2004, 15(09): 1385-1392.

[54] GOGUEN J, MESEGUER J. Security policies and security models[A]. IEEE Symposium on Security and Privacy[C]. Oakland, CA, 1982. 11-20.

[55] LATTNER C, ADVE V. LLVM: a compilation framework for lifelong program analysis & transformation[A]. 2004 International Symposium on Code Generation and Optimization[C]. San Jose, CA, 2004. 75-86.

[56] 曾海涛. 安全实时数据库隐蔽信道度量和处理技术研究[D]. 北京; 中国科学院研究生院, 2008.
ZENG H T. Research on Covert Channel Measurement and Handling in Secure Real-time Database[D]. BeiJing; Graduate School of the Chinese Academy of Sciences, 2008.

[57] BELL D, LA PADULA L. Secure Computer Systems: Mathematical Foundations[R]. MITRE CORP, 1973.

[58] SON S H, MUKKAMALA R, DAVID R. Integrating security and real-time requirements using covert channel capacity[J]. IEEE Transactions on Knowledge and Data Engineering. 2000, 12(6): 865-879.

[59] KEEFE T F, TSAI W T, SRIVASTAVA J. Database concurrency control in multilevel secure database management systems[J]. IEEE Transactions on Knowledge and Data Engineering, 1993, 5(6): 1039-1055.

[60] NEWMAN R C. Covert computer and network communications[A]. Proceedings of the 4th Annual Conference on Information Security Curriculum Development[A]. Kennesaw, Georgia. 2007. 1-8.

[61] PETITCOLAS F A P, ANDERSON R J, KUHN M G. Information hiding-a survey[J]. Proceedings of the IEEE, Special Issue on Protection of Multimedia Content. 1999. 1062-1078.

[62] HANDEL T, SANDFORD M. Hiding data in the OSI network model[M]. Information Hiding, 1996. 23-38.

[63] AHSAN K, KUNDUR D. Practical data hiding in TCP/IP[A]. Proceedings of ACM Workshop on Multimedia Security[C]. College Station, Texas, 2002. 1-8.

[64] CABUK S, BRODLEY C E, SHIELDS C. IP covert channel detection[J]. ACM Trans Inf Syst Secur, 2009, 12(4): 1-29.

[65] CABUK S, BRODLEY C E, SHIELDS C. IP covert timing channels: design and detection[A]. Proceedings of the 11th ACM conference on Computer and communications security[C]. Washington DC, USA. 2004. 178-87.

[66] AMIRUZZAMAN M, PEYRAVI H, ABDULLAH-AL-WADUD M, et al. Concurrent covert communication channels[A]. AST/UCMA/ISA/ACN'10 Proceedings of the 2010 International Conference on Advances in Computer Science and Information Technology[C]. 2010. 203-213.

[67] BERK V, GIANIA, CYBENKO G, et al. Detection of Covert Channel Encoding in Network Packet Delays[R]. Department of Computer Science, Dartmouth College, Technical Report TR2005536, 2005,

[68] GIANVECCHIO S, WANG H. Detecting covert timing channels: an entropy-based approach[A]. Proceedings of the 14th ACM Conference on Computer and Communications Security[C]. Alexandria, Virginia, USA. 2007: 307-316.

[69] YAO L, ZIX, PAN L, et al. A study of on/off timing channel based on packet delay distribution[J]. Computers & Security, 2009, 28(8): 785-794.

[70] LUO X P, CHAN E W W, CHANG R K C. TCP covert timing channels: design and detection[A]. International Conference on Dependable Systems & Networks[C]. Alaska, USA, 2008. 420-429.

[71] GIRLING C G. Covert Channels in LAN's[J]. IEEE Transactions on Software Engineering. 1987, SE-13(2): 292-296.

[72] ROWLAND C. Covert channels in the TCP/IP protocol suite, First Monday[EB/OL]. http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/528/449, 1997-5-5.

[73] ZIX, YAO L, PAN L, et al. Implementing a passive network covert timing channel[J]. Computers & Security, 2010, 29(6): 686-696.

[74] SELLKE S H, CHIH-CHUN W, BAGCHI S, et al. TCP/IP timing channels: theory to implementation[A]. 2009 INFOCOM [C]. Brizal, 2009. 2204-2212.

[75] LIU Y, GHOSAL D, ARMKNECHT F, et al. Hide and seek in time robust covert timing channels[A]. ESORICS 2009[C]. LNCS 5789. 2009. 120-135.

[76] MARTIN K, MOSKOWITZ I. Noisy timing channels with binary inputs and outputs[J]. IH 2006, LNCS 4437[C]. 2007. 124-144.

[77] HUFFMAN D. A method for the construction of minimum-redundancy codes[J]. Resonance, 2006, 11(2): 91-99.

[78] CHISNALL D. The Definitive Guide to the Xen Hypervisor[M].

Prentice Hall Press, 2007.

[79] 石磊, 邹德清, 金海. Xen 虚拟化技术[M]. 武汉: 华中科技大学出版社, 2009.
SHI L, ZOU D Q, JIN H. Xen Virtualization Technology[M]. Wuhan: Huazhong University of Science & Technology Press, 2009.

[80] MATTHEWS J N, DOW E M, DESHANE T, et al. Running Xen: A Hands-On Guide to the Art of Virtualization[M]. Prentice Hall PTR, 2008.

[81] SON J, ALVES-FOSS J. A formal framework for real-time information flow analysis[J]. Computers & Security, 2009, 28(6): 421-432.

[82] LANOTTE R, MAGGIOLO-SCHETTINI A, TRONA A. Time and probability-based information flow analysis[J]. IEEE Transactions on Software Engineering. 2010, 36(5): 719-734.

[83] GILES J, HAJEK B. An information-theoretic and game-theoretic study of timing channels[J]. IEEE Transactions on information Theory, 2002, 48(9): 2455-2477.

[84] NAGATOU N, WATANABE T. Run-time detection of covert channels[A]. The First International Conference on Availability, Reliability and Security[C]. Austria, 2006. 577-584.

[85] HAMADOU S, SASSONE V, PALAMIDESSIC. Reconciling belief and vulnerability in information flow[A]. 2010 IEEE Symposium on Security and Privacy[C]. Oakland, CA, 2010. 79-92.

[86] MILLEN J K. Finite-state noiseless covert channels[A]. Proceedings of the Computer Security Foundations Workshop II[C]. Franconia, USA, 1989. 81-86.

[87] TSAI C-R, GLIGOR V D. A bandwidth computation model for covert storage channels and its applications[A]. IEEE Symposium on Security and Privacy[C]. Oakland, CA, 1988. 108-121.

[88] AHMED Q N, VRBSKY S V. Maintaining security and timeliness in real-time database system [J]. Journal of Systems and Software, 2002, 61(1): 15-29.

[89] MOSKOWITZ IS, KANG M H. Covert channels - here to stay?[A]. COMPASS '94[C]. 1994. 235-243.

[90] MANIFESTO O C. Open cloud manifesto[EB/OL]. www opencloud-manifesto org. 2009.

[91] LIU Q, WENG C, LIM, et al. An In-VM Measuring Framework for Increasing Virtual Machine Security in Clouds[J]. IEEE Security and Privacy, 2010, 8(6): 56-62.

[92] GARFINKEL T, PFAFF B, CHOW J, et al. Terra: a virtual machine-based platform for trusted computing[J]. SIGOPS Oper Syst Rev, 2003, 37(5): 193-206.

[93] AZAB A M, NING P, SEZER E C, et al. HIMA: a hypervisor-based integrity measurement agent[A]. Proceedings of the 2009 Annual Computer Security Applications Conference[C]. Hawaii, USA, 2009. 461-470.

[94] SESHADRIA, LUK M, QU N, et al. SecVisor: a tiny hypervisor to provide lifetime kernel code integrity for commodity OSes[J]. SIGOPS Oper Syst Rev, 2007, 41(6): 335-350.

[95] KIL C, SEZER E C, AZAB A M, et al. Remote attestation to dynamic system properties: Towards providing complete system integrity evidence[A]. DSN 2009[C]. Lisbon, Portugal, 2009. 115-124.

[96] LITTY L, ANDRES H, LIE D. Hypervisor support for identifying covertly executing binaries[A]. USENIX Security Symposium [C]. San Jose, CA, 2008. 243-258.

[97] REZAEIM, MOOSAVIN S, NEMATIH, et al. TCvisor: a hypervisor level secure storage[A]. Proceedings of the Internet Technology and Secured Transactions (ICITST), 2010 International Conference for[C]. London, 2010. 1-9.

[98] KLEIN G, ANDRONICK J, ELPHINSTONE K, et al. seL4: formal verification of an operating-system kernel[J]. Commun ACM, 2010, 53(6): 107-15.

[99] MCDERMOTT J, FREITAS L. Using formal methods for security in the Xenon project[A]. Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research[C]. Oak Ridge, Tennessee, 2010. 1-4.

[100] MCDERMOTT J, FREITAS L. A formal security policy for xenon[A]. Proceedings of the 6th ACM workshop on Formal methods in security engineering. Alexandria, Virginia[C]. USA, 2008. 43-52.

[101] JOHN M, JAMES K, MYONG K, et al. ADA471608[R], 2007.

[102] GARFINKEL T, ROSENBLUM M. A virtual machine introspection based architecture for intrusion detection[A]. Proceedings of Network and Distributed Systems Security Symposium [C]. San Diego, CA, 2003. 253-285.

作者简介：

吴敬征（1982-），男，河北唐山人，硕士，中国科学院软件研究所博士生，主要研究方向为隐蔽信道分析、云计算安全、虚拟化技术、安全操作系统和网络信息安全。

丁丽萍（1965-），女，山东青州人，博士，中国科学院软件研究所研究员，主要研究方向为计算机取证、信息安全、操作系统安全和软件工程。

王永吉（1962-），男，辽宁营口人，博士，中国科学院软件研究所研究员、博士生导师，主要研究方向为隐蔽信道、虚拟化技术、人工智能、实时系统、网络优化、软件工程、优化理论和控制论。