基于可信计算的终端安全体系结构研究与进展*)

刘威鹏[1] 胡 俊[1] 方艳湘[2] 沈昌祥[3]

(中国科学院研究生院信息安全国家重点实验室 北京 100039)[1]

(南开大学计算机学院 天津 300071)[2] (海军计算技术研究所 北京 100036)[3]

摘 要 基于可信计算的终端安全体系结构研究是当前信息安全领域研究的新方向。本文首先对可信计算的关键模块TPM以及若干关键技术进行了深入的分析,而后概述了几个典型的基于可信计算的终端安全体系结构,最后讨论了当前体系结构研究存在的问题和今后的研究方向。

关键词 可信计算,终端,安全体系结构

Research and Development on the Secure Architecture of Terminal Based on Trusted Computing

LIU Wei-Peng[1] HU Jun[1] FANG Yan-Xiang[2] CHEN Chang-Xiang[3]

(The State Key Laboratory of Information Security of GSUAS, Beijing 100039)[1]

(Computer Institute of NanKai University, Tianjin 300071)[2] (The Computing Technology Institute of Navy Army, Beijing 100036)[3]

Abstract Research on the secure architecture of terminal based-on Trusted Computing is one of the new direction in the field of information security. This paper deeply analyzes TPM(Trusted Platform Module) which is the key module of trusted computing and the crucial technology of Trusted Computing, then summarizes the several typical terminal secure architecture based-on Trusted Computing. At last discusses the problem and the future research direction.

Keywords Trusted computing, Terminal, Secure architecture

当前保护信息系统安全的主要手段是以共享信息资源为中心,基于已有安全知识所生成的特征库检测系统文件和网络数据,在外围对非法用户和越权访问进行封堵,以达到保护的目的。但随着技术的发展,各种攻击手段越来越高明,人们只能把外围的封堵越做越复杂,相应投入到开发,管理,维护的费用也越来越高,然而实际的收效却不尽如人意。可见,传统的安全手段暴露出的其局限性,已经不能适应当前信息系统的安全需求。

产生这种局面的根源在于不去控制安全问题发生的源头——终端。从组成信息系统的服务器、网络、终端三个层面上来看,现有的保护手段是逐层递减的,这说明人们往往把过多的注意力放在对服务器和网络的保护上,而忽略了对终端安全的保护,这恰恰是人们对待信息安全问题认识上的一个误区,显然是不合理的。终端往往是创建和存放重要数据的源头,而且绝大多数的攻击事件都是从终端发起的。如果信息系统中每一个使用者都是经过授权和认证的,并且其操作都是符合安全策略的规定,那么就不会产生攻击性的事件,就能保证整个信息系统的安全,因此只有立足于终端,从终端安全入手才能更好地解决整个信息系统的安全问题,才能构筑全面高效的安全防护系统[1]。

近年来,终端安全的思想正在逐渐被人们所重视,对其研究也备受关注。在此背景下,于1999年,包括Intel、Microsoft、IBM等在内的业界几家大公司成立了可信计算平台联盟(TCPA Trusted Computing Platform Alliance),并于2003年更名为可信计算组织(TCG Trusted Computing Group)。TCG提出的可信计算(TC Trusted Computing)概念,其思路就是从终端安全入手,定义未来终端上的一种可信计算环境,通过建立这种可信计算环境来提供各种安全操作功能,达到提高终端安全性的目的[2]。

可信计算尽管是以一种工业规范的形式提出,但其思想却具有普遍而又深远的意义,实际上它是对安全问题的本质回归,使人们将解决信息安全问题的思路转移到解决终端安全问题上来。但是值得注意的是,TCG提出的可信计算平台的体系结构相对简单,它仅仅是在硬件上引入了TPM模块,因此要在其提出的可信计算平台上建立完整的"可信计算环境"还需要诸多技术手段的支持,而其中最为关键的是操作系统对平台体系结构的支持。目前还没有真正意义上完整的成熟的可信计算平台的产品,并且其中的诸多关键技术,例如,支持可信计算平台的操作系统安全体系结构的研究仍然处于起步阶段。

基于可信计算的终端安全体系结构研究是信息安全领域的一个新方向[3],它应该是一个结合平台硬件、安全操作系统、应用系统的完整的平台系统。本文对基于可信计算的终端安全体系结构实现进行了分类,并对研究状况进行了综述,最后讨论了当前体系结构研究存在的问题和今后的研究方向,这对于可信计算的后续研究具有一定的指导意义。

1 可信计算平台模块

TCG从行为角度来定义可信,即:如果一个实体的行为总是以所期望的方式达到预期的目标,那么该实体就是可信

*)本项目受973重点基础研究发展规划项目信息与网络安全体系结构研究(G1999035801)支持。**刘威鹏** 博士研究生,研究方向:可信计算,安全操作系统;**胡 俊** 博士研究生,研究方向:可信计算、安全操作系统;**方艳湘** 博士研究生,研究方向:可信计算,安全体系结构;**沈昌祥** 中国工程院院士,博士生导师,研究方向:安全体系结构。

的。实体可以被认为是一个平台，或者是运行在一个平台上应用或服务。TCG 认为如果从一个初始的“信任根”出发，在平台计算环境的每一次转换时，这种信任可以通过传递的方式保持下去（信任串）不被破坏，那么平台上的计算环境就始终是可信的。在可信环境下的计算平台上的各种操作也是可信任的，不存在不被信任的实体，恶意代码，计算机病毒因为是不被信任的，所以也不可能被运行，平台本身的完整性得到保证，安全自然也得到了保证。

可信计算的核心是可信平台模块（TPM Trusted Platform Model），它就是初始的信任根。TPM 通过物理方式和计算机主板相连。在 TPM 中有各种密码部件，这些部件能够提供 RSA 签名，密钥生成（Key Generation），HASH 运算（SHA-1）以及随机数产生（RNG）等功能，并且在 TPM 内部还带有非易失性存储装置（Non-Volatile Storage Device）和易失性存储装置（Volatile Storage Device），非易失性存储装置能够存储一些不变的身份信息和秘密信息，如背书密钥（EK Endorsement Key）和存储根密钥（SRK Storage Root Key），而易失性存储装置主要存储一些动态变化信息。TCG 规范[4]中对 TPM 的内部的结构描述如图 1。

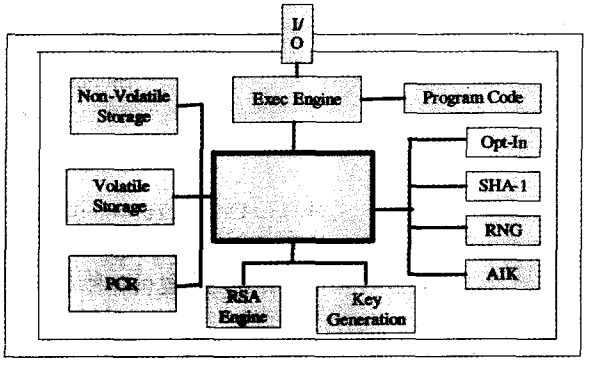

图 1　TPM 的内部结构

在 TPM 中，平台配置寄存器（PCR Platform Configuration Register）是一个非常关键的部件，它具有防窜改功能，主要用于存放可信串建立过程中的实体的度量值。出于安全和对于能够使用的 PCR 数量的考虑，每一次度量值是以如下扩展方式加入 PCR 的：

$[PCR(new)]=SHA-1([PCR(old)+(new-measured-value)])$即首先把事件数据和当前 PCR 中的值进行连接，而后计算连接值的摘要值，最后把该值存储在 PCR 中。

2　可信计算技术和可信计算平台

2.1　可信计算技术

TPM 是可信计算技术的核心，以 TPM 为信任根，TCG 的信任机制是通过可信度量（TM Trusted Measurement）、可信报告（TR Trusted Reporting）和可信存储（TS Trusted Storaging）来实现的。

可信度量：在可信串的建立过程中，任何将要转移控制权的实体（可信串的前一环），在控制权转移到下一个实体前（可信串的后一环）都必须对该实体进行度量，如果其符合某种要求，比如：该度量值和一个事先保存的预期值一致，则控制权才可以转移。在 TCG 针对的 PC 机的实施规范中，采用摘要计算作为可信度量的方式，而预期值也是预期摘要值，它是衡量实体是否按照“预期方式达到预期目标”的重要依据。通过可信度量，信任就可以从最初的可信根传递到操作系统，建立一个可信的操作系统环境，通过操作系统传递到应用系统，建立一个可信的应用环境，最后从应用传递到网络，建立一个可信的网络环境。

可信报告：对平台是否可信的验证和判定是通过可信报告的机制来实现的。任何想知道平台状态的是否可信的实体（通常称为挑战方）可以通过报告机制来获取 TPM 中度量信息和存储度量日志（SML Storage Measure Log），如果通过验证表明该度量信息满足某种特定的要求，则挑战方认为平台此时处于可信状态，否则认为平台的可信环境被破坏。

可信存储：通过 TPM 对重要的信息进行封装，TPM 本身的硬件特性就保证比放在其他硬件设备上安全得多，同时 TPM 又具有证明的能力，通过对存放的数据进行密封存储，可以更好地保护数据的完整性和机密性。

远程证明是可信计算最为重要的一个功能，它的主要目的是向远端的服务提供者证明系统当前软件环境的的状态，使服务的提供者相信系统行为是可信的，从而和本地系统进行交互。远程证明功能充分体现了可信计算的实质。远程证明的实现是以可信度量和可信报告机制为基础的。

基于 TPM 的存储的密封机制也是可信计算中另一个关键功能，其实质是：在某一时刻把所要求平台的配置值和某受保护的数据绑定，在另一时刻，只有平台所具有的配置值和所要求的配置相同时才能够获取该被绑定的数据。

此外可信计算中还包括其他硬件，软件，密钥和证书的管理诸多方面的内容，限于篇幅，不一一分析。

2.2　可信计算平台

如前所述，TCG 的目的是对各种终端平台提出符合可信计算设计思想的工业规范，因此考虑到平台的通用性，只定义了通用的参考结构[4]，如图 2 所示，其中包括了构建“可信根”的关键模块 TPM。

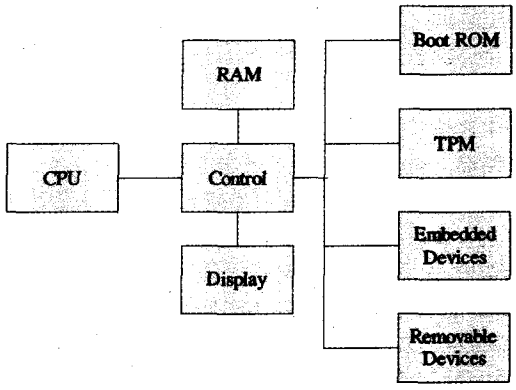

图 2　可信计算平台通用结构

为了降低硬件改动的成本以及和现有系统的兼容性，TCG 定义的可信计算平台结构仅仅增加了 TPM 模块，并且没有规定构建可信计算平台的软件系统，尤其是操作系统的设计。这也使得 TCG 所定义的可信计算平台本身并不是一个完整意义上的体系结构，从而触发了工业界和学术界对于基于可信计算的终端体系结构的研究。

3　基于可信计算的终端安全体系结构研究

TCG 提出的可信计算平台的体系结构相对简单，它仅仅在硬件上引入了 TPM 模块，因此要在其提出的可信计算平台上建立完整的“可信计算环境”还需要诸多技术手段的支

持，而其中的关键是操作系统的支持。而当前主流操作系统，如：Windows 操作系统和类 Unix 操作系统，在支持可信计算上主要存在以下的问题[6,7]：

1）没有良好的进程隔离机制；

2）缺少强制访问控制机制；

3）忽视最小特权策略；

4）没有为输入输出提供可信路径。

要克服主流操作系统存在以上的问题，必须站在体系结构的角度，研究以操作系统为中心的支持可信计算的终端安全体系结构。我们认为，在终端体安全系结构研究方面，可以归结为以下四个研究领域：基于安全内核的体系结构，基于微内核的体系结构，基于虚拟机的体系结构以及基于 LSM（Linux 安全模块）机制的体系结构。

3.1 基于安全内核的体系结构

安全内核是实现访问监督器的一种技术。通常，建立一个安全内核并不需要在它上面建立一个操作系统。安全内核可以很好实现操作系统的所有功能。设计者在安全内核中融入的操作系统的特点越多，安全内核就会变得越大。安全内核必须做得尽量地小。在设计时，必须坚决实施安全内核最小化地这个原则：凡不是维持安全策略所必需的功能都不应置于内核之中。进行安全内核设计时要考虑诸如性能、使用方便等因素，但这些于小型化要求相比，居于次要的地位。

采用安全内核构建终端安全体系结构的典型代表是 Microsoft 的 NGSCB，此外还有 George Mason 大学的 SecureBus 结构和由卡内基梅隆大学的 Bind 结构。

3.1.1 NGSCB 结构

NGSCB，即下一代安全计算基（Next-Generation Secure Computing Base），是 Microsoft 专有的适用于未来 Wndows 操作系统的硬件与软件相结合的平台体系结构[12]。在 NGSCB 中，可以在 PC 内部建立第二个作业环境，该环境用以保护系统免遭恶意程序代码入侵。作为保护的一部分，NGSCB 可以提供应用程序、周边硬件、内存以及内存之间的安全连接。

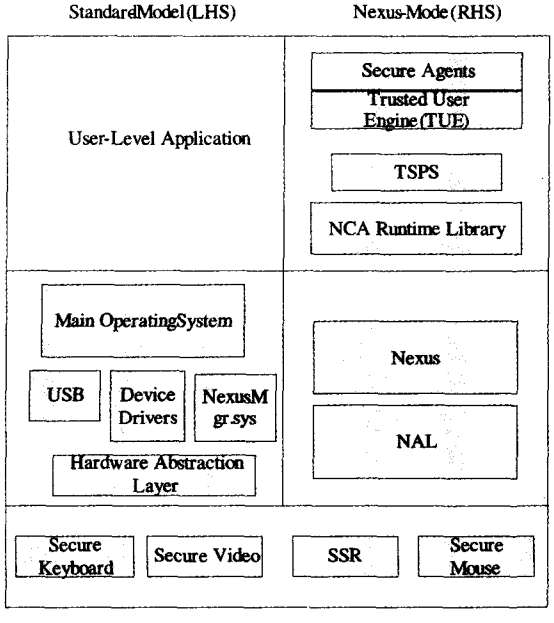

图 3 NGSCB 结构

NGSCB 需要新硬件的支持，包括 CPU、主板、TPM，以及其他专用芯片的支持，主要 Intel 命名该项计划的代号为 LaGrande 技术[13]，以提供密码、安全服务。这些新硬件的体系结构是向后兼容的，现有的操作系统和应用程序不需要改动就可在这些硬件平台上运行，但现有的应用程序若想使用 NGSCB 的安全服务则必须进行相应的改动。

NGSCB 为应用程序提供了一个受保护的操作环境（一个隔离的执行空间），在该环境中运行的程序不受主操作系统中恶意程序的任何影响，同时 NGSCB 也不会影响主操作系统中的程序。在支持 NGSCB 的计算机中，用户可以让现有的应用、服务和设备在不做任何改动的情况下运行在标准的操作系统环境中，而让关键的处理操作利用 NGSCB 服务在分离的、受保护的环境中进行。NGSCB 结构所提供的四项主要功能包括是：强进程隔离（Strong Process Isolation）；密封存储（Sealed Storage）；证明（Attestation）；用户安全路径（Secure Paths to the User）。图 3 给出了 NGSCB 体系结构的描述。

NGSCB 将系统分为四个部分，纵向的是用户和核心两级，横向的是标准和 Nexus 两个模式。标准模式就是现在通用的操作系统，如果计算机中没有安装或启用 NGSCB，则整个系统就只有标准模式。在启用 NGSCB 时，NexusMgr. sys 驱动程序将启动 Nexus 并与其中的服务进行交互。Nexus 是一个极小化的安全内核，它没有任何设备驱动及文件系统等。Nexus 管理器（NexusMgr）负责连接不安全的主操作系统（左侧）和安全的 Nexus（右侧），主要完成设备 I/O、文件系统、内存访问、用户接口等功能。Nexus 抽象层（NAL：Nexus Abstraction Layer）负责屏蔽底层硬件的差别，支持多处理器。Nexus 模式的用户级中包括 Nexus 计算代理（NCA：Nexus Computing Agents）、可信用户接口引擎（TUE：Trusted User Interface engine）、可信服务提供者（TSP：Trusted Service Providers）以及 NCA 运行时库，它们都用于各种安全程序的开发。

3.1.2 SecureBus 结构

SecureBus 是由 George Mason 大学的 ZhangXinWen 等人提出的一个可信计算体系结构[26]，如图 4 所示，该结构构建在可信计算技术硬件基础之上，SecureBus 结构在硬件上可以使用 TCG 的 TPM 和 Intel 的 LT 技术或 AMD 的 SEM 技术，并且添加了一个安全内核 SecureKernel（SK）和安全组件 SecureBus（SB）。该结构以 TPM 为可信根，而后依次把信任传递到 SK 和 SB 上。

SK 主要功能是结合底层硬件的功能对系统中进程和应用提供隔离功能。SB 位于操作系统内核空间和用户空间之间，主要功能是为进程分配独立的内存空间。此外，SB 还支持进程之间的灵活的强制访问控制策略，其访问控制结构在设计上采用了策略执行和策略决策相分离的框架，从而对于多策略和灵活策略提供了更好的支持框架。在原型系统实现时，主要实现了基于中国墙（Chinese-Wall）模型的访问控制机制。

该结构的主要特点是无需对原有的操作系统和应用进行修改，由于 SB 位于 SK 和上层应用之间，它对上层的应用和底层的操作系统来讲都是透明的。在应用层，它为上层应用提供了和原操作系统相同的系统调用接口，从而能够透明地执行真实性验证和访问控制策略。同时，由于 SB 代表受保护的应用来调用系统调用接口，从而无需要求对底层的操作系统进行修改。

该结构的另一个特点是能够保证进程和数据的真实性。

因为由底层硬件和SK所提供的隔离性质可以防止对出于运行状态的代码进行修改。但是对于保护代码的完整性是不充分的。因为攻击者可以采用对进程进行非法输入的方法来破坏完整性,所以在SecureBus中采用了对进程的输入和输出进行Hash运算的方法来解决,进程代码在加载之前被SB进行Hash运算。进程的输出数据被SB签名,并和进程代码和输入数据的Hash值连接在一起。

对于远程证明,基于可信根TPM,通过构建Hash串来建立对于SB和上层应用的信任。SK具有一个在平台初始化时由TPM所产生的公-私密钥对,并且该公钥由TPM的证明身份密钥(AIK)证明。SK也为SB产生一个公-私钥对,其中公钥由SK证明,其私钥由SB通过TPM所提供的密封存储功能进行保护。SB的公-私钥对由SB第一次在平台上安装时产生。为了对于一个正在运行的进程状态进行证明,TPM使用自身的AIK密钥签署一组平台寄存器的值,而SK使用自己的私有密钥签署SB的完整性值,而SB签署应用代码的完整性值。而后这三个签名被发送到挑战者方。而挑战者验证所有发送来的签名和以及AIK的公钥证书和SK以及SB的公钥证书。如果所有证书是有效的,并且完整性值能够匹配,那么该应用就是可信的。

此外,通过使用从平台可信根到SK并到应用的可信串传递,SB支持基于进程的证明,从而支持进程间可信的通信和协作。在单个平台上,SB作为一个被隔离进程的可信代理来和其他的进程进行通信。而在分布式网络环境中,平台上的SB和另一个平台上的SB通过远程证明来构建可信通道。

3.1.3 Bind结构

Bind是由卡内基梅隆大学的Elaine Shi等人提出的,用于为安全分布式系统的细粒度证明服务[25],其主要特性包括:

1)执行更细粒度的证明:Bind允许编程者标识一个进程,并注明需要证明的内容的开始到结束的部分,而不是证明整个内存的内容,通过这个证明注释(annotation)机制,可以简化Hash验证并解决软件版本升级的问题;

2)缩短了证明时间和使用时间之间的间隔:Bind在代码内容执行之前度量代码,即对代码进行Hash运算,而后使用沙箱机制对代码的执行过程进行保护,当代码在沙箱中执行完毕后,把代码的度量和代码所产生的数据进行绑定,从而能够查明是什么代码被运行来产生该数据;

3)通过把对输入数据的完整性验证整合到证明中,Bind提供了可传递的完整性验证;这是通过标识符(authenticator)和一些外部机制来保证的,例如,证书,语义检查和可信路径。

在Bind结构中,每一个平台上都装有可信硬件,包括TCG的TPM芯片和AMD的SEM处理器[26]。以TPM为信任根,而后通过TCG的安全启动和加载时间(Load-time)的证明来建立对于安全内核(SK)的信任。

Bind构建于SK中,其为普通应用提供证明初始化(attestation_init)和证明完成(attestation_complete)两接口:

1. Interface attestation_init
   In(input data memory address, size of process code)
   Out(success indicator)
2. Interface attestation_complete
   In(output data memory address)
   Out (authenticator)

其具体保护实现过程如下:首先进程被存储在一个相临的内存区域中。进程执行时,首先调用attestation_init()来初始化证明过程,attestation_init()的参数包括:进程输入数据的内存地址和进程代码的大小。在接收到attestation_init()请求后,Bind首先验证进程输入数据的鉴别符Authenticator,该鉴别符声明了:该数据是由一个合法的进程运行真正的输入数据产生的,而后进程通过SK对进程代码和输入数据地址进行Hash计算。为了确保Hash计算的内容和所执行内容的一致,Bind通过沙箱机制对执行的进程建立隔离环境。以上成功完成后,Bind将会把控制权转交给进程,并为其标记成功标识符(success indicator)。此后直到attestation_complete()命令完成,该进程都能够确保在一个受保护的环境中安全地执行。当进程执行完毕调用attestation_complete()命令后,Bind对输出数据和进程代码计算鉴别标签(tag),该tag把输出数据和产生该上面计算的输出数据的代码的Hash值绑定在一起。而后Bind解除对于该对进程的保护,并把鉴别符(Authenticator)返回给进程。

即使有强的隔离机制对于运行完整性的保证,然而并不能够保证(1)通信方的真实性以及应用之间的输入和输出;(2)应用之间灵活的访问控制机制。而这两方面对于保证这个系统的完整性是非常重要的,而不是仅仅依赖于代码的完整性。在(1)中,进程需要确保所接收到的数据是真实的。在(2)中,即使源进程是可信的,它可能具有更低的完整性或机密性。因此需要在进程之间执行访问控制来满足系统安全管理员所定义的某些安全需求。这也是SecureBus和Bind结构给我们开发安全体系结构一个启发。

值得注意的是,尽管Bind结构也使用Hash串来对进程以及进程的数据进行完整性检验,但是SecureBus和Bind所采用的方法是不同的。首先,在SecureBus结构中,进程代码以及进程的输入和输出值都包括在签名中,并且一起被发送给下一个接受进程。这不仅仅是进程代码的完整性。而实际上是需要验证进程的输出来自于进程的输入,因为在实际中验证进程的真实的输出来自于真实的输入是非常必要的。而在Bind的结构中,仅仅验证进程代码以及进程的输出,这就无法保证进程的输入的真实性。其次,在SecureBus结构中,完整性验证是由平台上的SB来完成的,而这个功能对于应用来讲是透明的,然而在Bind结构中,出于细粒度证明的需要,完整性度量和证明是基于一个进程的关键的部分,并且需要通过调用BIND所提供的相应的功能来完成的。因此完整性验证是通过单个应用来执行完成的,这将使得安全功能对于应用来讲并不是透明的。

3.2 基于微内核的结构

微内核(Microkernel)是和单(宏)内核(Monolithic kernel)相对的概念,在微内核中,大部分模块都是独立的进程,并在一定的特权状态下运行,各个模块通过消息传递的方式进行通信。单内核无论从结构的安全强度,对进程的隔离功能,对于安全策略支持的灵活性以及扩展性方面以及对应用的支持方面都不如微内核[37],这也是选择微内核构建终端安全体系结构的一个重要原因所在。

3.2.1 EMSCB结构

EMSCB,即欧洲多边安全计算基(European Multilateral Secure Computing Base)[14],该项目是针对Microsoft的NGSCB而发起的,其目的就是提供面向开源的可信计算平台架构,解决传统计算平台中的一些安全问题。EMSCB主要优点包括:一方面支持多边安全,即提供给用户对于恶意代码良

好的保护机制，防止违反用户的安全策略，另一方面，保护内容的提供者防止用户对条约的破坏。EMSCB 具有：开放的结构，可信赖地使用 TCG 的技术，低价格的可携带性，以及未来的确保性等特点。EMSCB 的结构如图 4 所示。

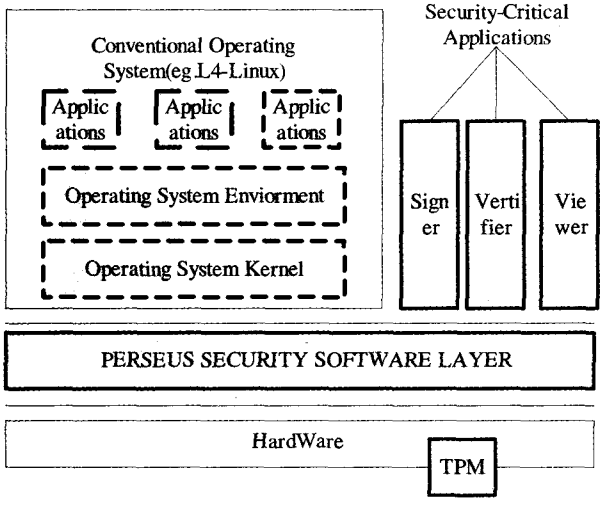

图 4　EMSCB 体系结构

在硬件层主要添加了 TCG 的 TPM 模块，来完成远程验证底层硬件的完整性；把平台的配置和秘密信息绑定；以及安全存储密码密钥等功能。在硬件层之上是基于微内核的开源安全内核 PERSEUS[15,16]，它是一个非常小的微内核，能够控制底层的安全硬件来保护安全相关的应用和敏感的信息。PERSEUS 并没有虚拟底层的硬件接口，而是为上层应用提供了一个更为抽象的接口来对操作系统服务。在 PERSEUS 之上，是一个经过修改的通用的操作系统（例如 Linux），它是作为一个任务来执行，并受到 PERSEUS 的控制和保护。和操作系统并列的是一系列新增添的安全应用（例如，数字版权管理 DRM），这些应用能够使用可信计算提供的新的特性。EMSCB 通过使用由 CPU 提供内存保护机制和 PERSEUS 提供的安全的进程间通信机制（IPC Inter-Process Communication），能够很好地在这些应用之间以及应用和操作系统之间实施隔离。

3.3　基于虚拟机的结构

微内核和虚拟机监视器（VMM Virtual Machine Monitor）都是操作系统研究中的重要领域。在实施隔离机制方面，微内核和虚拟机都表现出色，它们有一些非常重要的特点：二者都将系统划分成隔离的，通过精确定义过的（通常来说也是严格控制的）接口进行通信的组件，而且结构上也有很多相似之处。二者的主要区别是微内核在上世纪八九十年代在学术界获得了很多关注而虚拟机监视器的研究主要是工业领域。虚拟机监视器另外一个重要机制是硬件分享，为多个虚拟机提供在单一硬件平台上安全的多路复用。虚拟机管理器在用户与硬件之间增加了一个中间层，因此需要确保它在性能上的损失要尽量小。人们在这方面进行了大量研究，也取得了很大的成果。尽管虚拟机管理器体系结构对客户操作系统所需改动的要求程度不同，但基本都只需要很小的改动甚至不改动。

3.3.1　Terra 结构

Terra 是一个基于 VMM 的简单而又具有灵活性的模型[17,18]。它允许开发者根据具体应用需求定制操作系统——甚至是不需要操作系统，就像在一个封闭式专用硬件平台上那样。对于其上的操作系统和应用程序来说，Terra 就是一个真实的硬件平台。

TCG 对 Terra 提供充分的硬件支持，Terra 可以使用现有 TCG 提供的防篡改的硬件设备 TPM 和各种安全功能：利用 TCG 提供的可信启动验证系统启动过程；需要 TCG 提供的密封存储功能存储各可信操作系统的摘要值用于比对等。图 5 是 Terra 的结构。

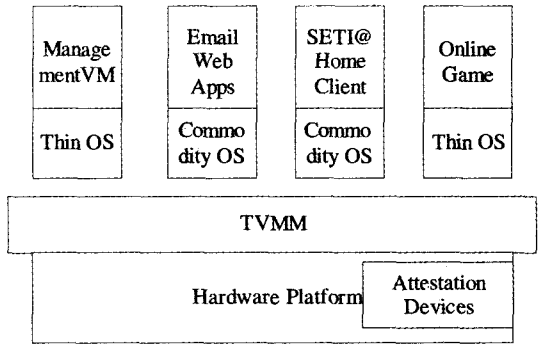

图 5　Terra 核心部件——TVMM 结构

Terra 的核心是可信虚拟机监视器（TVMM Trusted Virtual Machine Monitor），它在一个高可靠性的防篡改的通用平台上向用户提供多个互相隔离的虚拟机的底层部件。TVMM 具有 VMM 的功能：隔离性、扩展性和兼容性。此外它还具有三个附加的对实现封闭式 VM 很重要的安全功能：

1）具有安全根，平台所有者也不能破坏，以便 TVMM 为封闭式 VM 提供的私密和独立性；

2）TVMM 作为可信方可以对远端对象认证 VM 中运行的软件；

3）提供用户和应用之间的可信路径。

通过 TVMM，Terra 达到了既支持面向封闭式专用计算环境的应用系统（通过封闭式 VM），又支持现有通用操作系统和应用（通过开放式 VM）的目标。

TVMM 负责保护封闭式 VM 的私密性和完整性，它通过硬件内存保护和存储加密保护手段实现了封闭式 VM 和系统其它组件的隔离，它还允许采用类似 TCG 规范中的远程证明机制向远端认证自身完整性。可信通道通过给用户提供安全的、不可绕过和不可更改的用户界面及明确的当前所使用的 VM 标识来实现，不过真正实现这个要求需要显示设备在硬件和软件上的支持，目前在原型 Terra 系统上并未实现。

Terra 没有实现完整的可信度量体系，各个虚拟机需要实现属于本虚拟机上的度量机制，而 TVMM 只能算是一个软件安全根。Terra 中对驱动程序的隔离和保护考虑得较少，因此应用需要全部信任设备驱动。另外，Terra 虚拟机之间缺少 IPC 机制，不利于做更多的扩展。更重要的是，目前 Terra 还不是开源系统，因此并非一个合适的研究平台。

和 Terra 相比，Xen VMM[19~21] 是由剑桥大学计算机实验室开发的一个开源项目，Xen 可以支持创建多个的虚拟机，每一个虚拟机都是运行在同一个操作系统上的实例。Xen 是一款半虚拟化的（Paravirtualizing）VMM，这表明为了调用系统管理程序，要有选择地修改操作系统（已经对 Linux、Opensolaris、FreeBSD 作了修改，对 WindwsXP 的移植还没有完全完成），但是不需要修改操作系统上运行的应用程序。

在 Xen 系统中，主域与主域之间，主域与 Xen Hypervisor 之间的接口也极为简单，而且允许设备和操作系统对接口进

行扩展，同时它的设备结构出于可靠性目的使驱动程序被隔离在一个虚拟机中，低层接口具有扩展性而不必修改客户操作系统或虚拟机。

Xen具有成熟和稳定的特性，Xen系统上可以很方便地实现模块化(类似微内核系统)和功能扩展。此外Xen还具有灵活性，可以根据需要灵活配置主域的资源。客户操作系统作为虚拟机结构中的模块，它的大小也可以根据需要进行裁减。

如何在Xen上构建终端安全体系结构是一个值得研究的热点。

3.4 基于LSM机制的结构

以上分别采用了安全内核、微内核和虚拟机三种结构，这可以说是对于原始的终端体系结构进行本质上的改变，而通过采用LSM机制，则是在原有主流操作系统上的安全增强，使其对于可信计算进行很好的支持。LSM是Linux安全模块(Linux Secure Module)的简称，它是Linux内核的一个轻量级的、通用的访问控制框架，它允许多种不同的访问控制模型作为可加载内核模块实施，用户可以根据其需求选择合适的安全模型加载到Linux内核，该框架满足：

1)真正通用，使得采用不同的安全模型仅仅是加载不同的内核模块；

2)概念简单，高效，而且对内核影响小；

3)支持现有的POSIX.1e权能逻辑，将其作为一种可选的安全模块。

LSM的基本思想就是在内核的数据结构中放置钩子函数(Hook Function)，由钩子函数对内核对象的访问控制权限作出判断。使用LSM进制来构建基于可信计算的终端安全体系结构具有易用性和灵活性的特点。使用LSM机制实现终端安全体系结构的典型代表是由IBM开发的完整性度量结构IMA，PRIMA是对于IMA存在问题的一个扩展，而TLC中应用了IMA。

3.4.1 Bear结构

Bear是Dartmouth大学的一个开源项目[23,24]，其目的是希望借助TCG的安全硬件TPM解决现实系统中的安全问题。Bear由4部分组成，包括：Enforcer模块、Privacy CA、TPM-enabled LILO和Security Admin应用工具，而其中的Enforcer模块是Bear中的主要部分。

Enforcer包括一个TPM驱动模块和一个LSM访问控制实施模块。Enforcer属于常驻内核的一部分，在具体实现时采用了LSM钩子方式。整个Enforcer的运行分为两个部分：第一部分是初始化部分，在启动时检查一个被签名的配置文件，运行另外的合适任务。第二部分是运行部分，检查在medium-lived配置中文件的完整性。

TPM通过保护PCR来保护Enforcer，Enforcer则通过LSM、OS、回绕密钥来保护短期的应用程序、通过对比被签名的配置和Security Admin中的Public Key相比较，来保护长期存在的应用程序。而客户端可以根据Enforcer的比较情况，来判断该服务是否可信。

Enforcer的完整性校验机制将保护对象分成三种类型："短寿命数据"(例如网页文件等)，"中寿命软件"(例如APACHE等)，"长寿命核心"(例如内核、模块等)。

1)对"短寿命数据"，将它们放到encrypted loopback file-system(加密的本地回环设备)中去；

2)对"中寿命软件"，对于中期存在的数据，我们需要一种方法来使得远程用户来识别系统的安全相关配置，以及一个工具来检查是否该系统合该配置相匹配。

3)对"长寿命组件(内核及内核态组件)"，它们都由TPM中的PCR直接验证：BIOS，OS LOADER，内核，Enforcer模块，Security Admin的公钥。Enforcer采用LSM模块的方法实现。

Security Admin中定义了必要的安全策略用于实施安全判定。Enforcer采取/etc/enforcer/来存储签名密钥、公钥等。在内核初始化Enforcer时，Enforcer在LSM框架注册hook，在启动时间里，Enforcer将会检查被签名的安全策略，如果该安全策略已经编译到内核中，Enforcer将会在根文件系统被挂载时，对其实施验证。在运行时，Enforcer对所有的节点实施允许检查。Enforcer计算节点文件的SHA-1值，同时将其和策略中的SHA-1值比较。如果该值不匹配的话，则根据具体的选择来进行实施，把事件记录到系统日志中，使该调用失败，使系统处于Panic状态。

Bear不是完全按照TCG规范设计的，只是部分利用了TPM的功能，而且没有完整的可信度量机制。虽然，Enforcer采用了Linux操作系统的LSM机制，但Linux系统固有的隔离性差问题仍然存在，在Enforcer基础上很难实现真正的可信计算。

3.4.2 IMA结构

IMA(Integrity Measure Archecture)是IBM的Reiner Sailer等人开发的一个最早的著名的基于TCG的可扩展的完整性度量结构[8~10]。IMA在Linux系统(Redhat9.0 kernel2.4.21)上实现，通过对系统中的可执行文件、动态加载器、内核模块以及动态库进行度量来保证系统运行时间(Runtime)的完整性。所有的度量值采用TPM_Extend操作扩展到TPM的PCR中，同时在系统内核中维持一张有序的度量列表ML(Measure list)。当挑战方(challenger)需要验证证明方(attestor)状态时，采用一个防止证明信息被重放，窜改和伪造的完整性挑战协议。该挑战协议结束时，挑战方获得了证明方的PCR值和ML，而后通过ML计算虚拟的PCR值，把虚拟的PCR值和挑战方发送的PCR值进行对比，在两者一致的情况下，就可以确定该ML是真实未窜改的。在挑战方自身需要维护的一个安全数据库，用来存放正确的组件度量信息值，然后可以使用该数据库来验证证明方的状态是否可信。

IMA实现主要包括三个主要部分：在系统中插入度量点，实际度量和对度量值的验证。其中插入度量点是实现的重点，主要采用了基于Linu的LSM钩子机制[11]，主要在系统调用insmod()、execve()、动态加载器和脚本解释器(perl脚本解释器)中添加了度量函数measure()。

Reiner等人的研究成果表明：该原型系统是可以扩展的，许多在Microsoft的NGSCB中实现的功能，不需要新的CPU模式或是对操作系统很大的改进，就在现今的硬件和软件系统上实现，而仅仅依赖于使用TPM。

3.4.3 PRIMA结构

PRIMA是由IBM的Trent Jaeger等人对IMA扩展的基础上提出的一个策略减弱的完整性度量结构(Policy-Reduced Integrity Measure Architecture)[32]，是对IMA的一个扩展和增强，解决了在IMA结构中存在的加载时间(Load-time)度量问题和过多度量问题。加载时间度量问题是在体系结构设计中引起研究者关注并经常被讨论的问题，其是指在代码(包

括:可执行文件、动态加载器、内核模块、动态库等)加载时间度量仅仅能够确保代码在被加载的时间是高完整性的,即表明能够导致代码注射攻击的信息流已经被程序充分处理,从而在此上下中,高完整性程序是一个不具有已知脆弱性的程序。然而这样的程序它所依赖的数据如果被恶意地修改或者程序所依赖的数据来源于一个不可信的输入影响,那么先前的不可知的脆弱性就可能会被利用。

加载时间度量,在关于哪些代码必须是可信方面会导致更为保守保证,因为在实际当中,仅仅是那些目标应用所依赖的代码和数据需要是高完整性的。而如果像 IMA 那样,对于系统中所有程序(脚本,库)都进行度量,就会产生过度度量问题,它将会直接影响度量结构的效率。

PRIMA 是建立在许多基础性的工作[29-31]之上的,通过使用一个减弱的 Clark-Wilson 模型:CW-Lite[29],来防止低完整性的输入数据破坏应用的完整性,以及过多度量问题。和 IMA 相比,PRIMA 在完整性度量时,除了对原有的代码和静态数据的基本度量外,还需要以下的度量:

1)MAC 策略:MAC 策略确定了系统的信息流,因此是必须要进行度量的;

2)可信的主体:和目标应用相交互的可信主体的集合要被度量,远程方也必须要对这个集和中仅仅包括它信任的主体达成一致;

3)代码——主体映射:对于所有被度量的代码,需要记录下代码和加载该代码的主体之间的映射关系。

PRIMA 的实现思路如下:在系统启动时,MAC 策略和可信的主体集被度量。通过这些度量,远程方能够构建一个信息流图。远程方能够验证所有来自于可信主体(该主体在运行时间被验证运行着可信的代码)或具有经过过滤器接口过滤过的来自于非可信主体所有流行目标应用和可信应用信息。而后,度量运行时间的信息。根据信息流图,仅需要度量所需要依赖的代码。其他代码都假定为不可信的。而后还需要度量在加载代码和加载该代码主体之间的映射,从而远程方能够验证该主体执行了预期的代码。

PRIMA 仅仅要求附加地度量 MAC 策略和在加载时间的可信主体,以及代码和 MAC 策略主体之间的匹配问题,由于不再需要度量不可信的主体,从而就可以减少一部分度量值。PRIMA 结构充分体现了安全操作系统对于可信应用的支持上的典型工作。

3.4.4 TLC 结构

IBM Watson 研究中心基于 Linux2.6 内核,使用 LSM 机制和 TPM 硬件模块开发了可信 Linux 客户端 TLC(Trusted Linux Client)[34]。TLC 通过 TPM 实现引导时完整性测量,可装载内核模块 EVM(Extended Verification Module)验证运行时系统中所有文件的完整性,SLIM(Simple Linux Integrity Module)通过实施强制访问控制保护系统文件的安全性和完整性。TLC 主要使用 TPM 的封装存储功能,保护内核主密钥的安全,同时提供安全引导,保证内核(包含 EVM 模块)的完整性。TLC 中的所有文件都定义了认证扩展属性,存储文件完整性验证所需信息和其安全属性(保密级和完整级),每个文件第一次打开或执行时,EVM 检查其完整性。SLIM 根据 EVM 的完整性校验结果,实施强制访问控制,验证可信的进程可以在权限范围内进行访问。若验证为不可信进程,则 SLIM 控制该不可信进程只能访问部分资源。TLC 中扩展安全属性验证文件完整性的方法,仅限于本机内部所有文件,它无法解决网络中远程证明的问题。

4 当前研究存在的问题和下一步研究方向

在上述的概述典型的终端体系结构中(除 Bear,IMA,TLC 外),尽管所使用的底层的硬件和体系结构的构建结构的方式不同,但是都把对于应用或进程的隔离作为其主要实现的目标之一。在这些体系结构上,已有的隔离技术主要是通过以下三种方法实现的:

1)通过使用强制访问控制机制对于应用进行隔离,例如 PRIMA 结构;

2)通过使用虚拟机器的技术,例如 Terra 结构和 Xen 结构;

3)通过使用底层增强的安全硬件如:Intel 的 LT 技术或 AMD 的 SEM 技术,并且可以添加安全内核或者是微内核,例如 NGSCB 结构,EMSCB 结构,Bind 结构和 SecuereBus 结构。

这就为基于可信计算的终端体系结构的构建提供了很好启示,即体系结构设计时应该把对于应用或进程的隔离作为一个关键目标。

但是我们同时也认为,隔离只是做好可信计算的必要条件,是关键目标之一。在系统中,进程之间的通信是必然的,正如 XinWenZhang[26] 和 Trent Jaeger 等人所指出的,通过进程的隔离保证进程的安全是不充分的,还需要对于进程的输入和输出以及进程之间的信息流进行必要度量和控制,这也是当前研究领域所关心的热点问题。

在对于安全体系结构研究进展的概述时,我们指出微内核和虚拟机监视器都是操作系统研究中的重要领域。微内核和虚拟机都可以实施细粒度的隔离,相对于微内核系统来说,虚拟机有优势在于:虚拟机监视器在结构上更为合理,具有良好的兼容性以及性能较高。而跟虚拟机监视器相比,微内核结构有三个主要特性体现在:系统模块之间定义接口简单精炼以利于设备和操作系统功能的扩展,相对大内核来说具有更少的核心代码,利于进行系统安全性的验证以及组件间隔离性好,利于提高可管理性。

因此在实际中构建安全体系结构时,应该根据实际的需求,可以在两者中间作出合理的选择。

采用 LSM 框架开发终端安全体系结构具有很好的适用性和灵活性,因此被广泛地采用。但是存在以下不足:仅仅采用 LSM 机制通过添加安全钩子的方式是无法提供应用或进程之间的隔离。在文[37]中,也对于 IMA 的工作进行了评述:"Sailer 声称:通过使用现有的硬件和软件,能够获取许多 NGSCB 的保证,并且不要求新的 CPU 模式和操作系统,而仅仅依赖于独立的可信实体:TPM。这样的结构完成一部分保证是可能的,然而,最为重要的保证(密封存储和远程证明)是难以取得,因为可信系统的关键组件必须被操作系统和基于硬件 CPU 的保护机制结构的联合使用,而仅仅依赖于增加硬件(TPM)是不行的。"我们认为在主流操作系统上仅仅通过 LSM 机制实现实际的度量是不足以支持可信计算的,有效的域隔离,强制访问控制机制和最小特权是在主流操作系统上支撑可信计算的根本要求。

结束语 本文对基于可信计算的终端安全体系结构的研究和进展进行了研究综述,包括 TPM 结构和功能,可信计算的关键技术,并对基于可信计算的终端安全体系结构的研究

(下转封四)

(上接第 263 页)

进行了分析和概述,并指出研究的关键是支持可信计算体系结构的安全操作系统的研究,最后讨论了当前研究存在的问题和今后的研究方向。通过本文的研究,使我们能够对基于可信计算终端安全体系结构的研究现状有了更为清楚和直观的认识,从而能够更好地把握可信计算技术给信息系统安全带来全新的安全理念和使用模式。

参 考 文 献

1 沈昌祥.基于积极防御的安全保障框架.中国信息导报,2003(10)
2 http://www.trustedcomputinggroup.org
3 沈昌祥.可信计算平台和安全操作系统.网络安全技术与应用,2005.4
4 TCG Specification Architecture Overview, Version1. 2. https://www.trustedcomputinggroup.org
5 TCPA Main Specification v1. 1b TCPA[EB/OL]. https://www.trustedcomputinggroup.org/home 2003-09-02
6 Jason F, Reid William J, Caelli D R M . Trusted Computing and Operating System, In: Architecture Australasian Information Security Workshop, 2005 (AISW2005)
7 Jason Reid Juan M, Gonz'alez Nieto, ed. Dawson Privacy and Trusted Computing. In: Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03)
8 Sailer R, Zhang X, Jaeger T, van Doorn L. Design and implementation of a TCG-based integrity measurement architecture. In: Proceedings of the 11th USENIX Security Symposium. USENIX, Aug. 2004
9 Sailer R, Van Doorn L, James P. Ward The Role of TPM in Enterprise Security: [IBM Research Report RC23363 (W0410-029)]. October 2004
10 Integrity Measurement Architecture (IMA) 2006-4-18. http://domino.reaearch.ibm.com/comn/research_projects.nsf/pages/ssd_ima.index.html
11 Wright C, Cowan C, Morris J, et al. Linux security modules: General security support for the Linux kernel. In: Proc. of the 11th USENIX Security Symp. San Francisco, 2002. 17~31. http://www. usenix. org/events/sec02/full_papers/wright/wright_html
12 Microsoft. Microsoft Next-Generation Secure Computing Base - Technical FAQ. February, 2003. URL: http://www.microsoft.com/technet/treeview/default. asp? url =/technet/security/news/NGSCB.asp (May 30, 2003)
13 LaGrande Technology Architectural Overview 252491-001 September 2003. http://www.intel.com/technology/security/downloads/LT_overview_fail_idf13.htm
14 Sadeghi A-R, Stüble C. Norbert Pohlmann European Multilateral Secure Computing Base
15 Pfitzmann B, Riordan J, Stüble C, Waidner M, Weber A. The PERSEUS System Architecture: [IM Technical Report #93381]. IBM Research Division, Zürich, 2001
16 Sadeghi A-R, Taming C S. Trusted Computing by Operating System Design. In: Proceedings of the 4th International Workshop on Information Security Applications (WISA), Korea, 2003
17 Garfinkel T, Rosenblum M, Boneh D. Flexible OS support and applications for trusted computing. In HotOS-IX, 2003
18 Garfinkel T, Pfaff B, Chow J, Rosenblum M, Boneh D. Terra: a virtual machine-based platform for trusted computing. In: Proceedings of the nineteenth ACM symposium on Operating systems principles, ACM Press, 2003. 193~206
19 Clark T, Deshane E, Dow, Evanchik S, Finlayson M, Herne J, Matthews J N. Xen and the art of repeated research. In: Proceedings of the Usenix annual technical conference, Freenix track, July 2004
20 Dragovic, Fraser K, Hand S, Harris T, Ho A, Pratt I, Warfield A, Barham P, Neugebauer R. Xen and the art of virtualization. In: Proceedings of the ACM Symposium on Operating Systems Principles, October 2003
21 Enforcer Project Homepage. http://enforcer.sourceforge.net/, March 2005
22 MacDonald R, Smith S, MarchesiniJ, Wild O. Bear: An open-source virtual secure coprocessor based on TCPA: [Tech. Rep. TR2003-471]. Department of Computer Science, Dartmouth College, 2003
23 Enforcer Project Homepage. http://enforcer.sourceforge.net/, March 2005
24 MacDonald R, Smith S, Marchesini J, Wild O. Bear: An open-source virtual secure coprocessor based on TCPA: [Tech. Rep. TR2003-471]. Department of Computer Science, Dartmouth College, 2003
25 Shi E, Perrig A, Van Doorn L. Bind: a fine-grained attestation service for secure distributed systems. In: Proceeedings of IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 2005. 154~168
26 AMD Platform for Trustworthy Computing Windows Hardware Engineering Conference 2003
27 Zhang Xinwen, Chen Songqing, Michael J. Covington, and Ravi Sandhu SecureBus: Towards Application-Transparent Trusted Computing with Mandatory Access Control
28 Jaeger T, Sailer R, Shankar U. PRIMA: Policy-Reduced Integrity Measurement Architecture
29 Shankar U, Jaeger T, Sailer R. Toward automated information-flow integrity for security-critical applications. In: Proceedings of the 13th Annual Network and Distributed Systems Security Symposium. Internet Society, 2006
30 Jaeger T, Sailer R, Zhang X. Analyzing integrity protection in the SELinux example policy. In: Proceedings of the 12th USENIX Security Symposium, USENIX, August 2003. 59~74
31 Jaeger T, Sailer R, Zhang X. Resolving constraint conficts. In: SACMAT '04: Proceedings of the ninth ACM symposiumon Access control models and technologies, New York, NY, USA, ACM Press, 2004. 105~114
32 Jaeger T, Sailer R. Umesh Shankar PRIMA: Policy-Reduced Integrity Measure Architecture
33 Yoshihama S, Ebringer T, Nakamura M, Munetoh S. Hiroshi Maruyama WS-Attestation: Efficient and Fine-Grained Remote Attestation on Web Services. In: Proceedings of the IEEE International Conference on Web Services (ICWS'05)
34 Safford D, Zohar M. A trusted Linux Client. IBM T. J. Watson Research Center, 2004
35 陈幼雷,等.操作系统可信增强框架研究与实现.计算机应用与研究
36 林宣雄,等.安全内核方法与实现考虑.计算机科学,1996(11)
37 Reid J F, Caelli W J. DRM, Trusted Computing and Operating System Architecture. In: Australasian Information Security Workshop. (AISW2005), Newcastle, Australia

计算机科学

(1974 年 1 月创刊)

第 34 卷第 10 期 (月刊)

2007 年 10 月 25 日出版

国际标准连续出版物号 ISSN 1002-137X

国内统一连续出版物号 CN50—1075/TP

定价: 30.00 元 国外定价: 5 美元

邮发代号: 78—68

发行范围: 国内外公开

主管单位: 国家科学技术部

主办单位: 国家科技部西南信息中心

编辑出版:《计算机科学》杂志社

重庆市渝北区北部新区洪湖西路 18 号 邮政编码: 401121

电话: (023) 63500828 E-mail: jsjkx@swic.ac.cn

网址: www.jsjkx.com

社 长: 牟炳林

总 编: 彭 丹

主 编: 朱宗元

主编助理: 徐书令

印 刷 者: 重庆科情印务有限公司

总发行处: 重庆市邮政局

订 购 处: 全国各地邮政局

国外总发行: 中国国际图书贸易总公司 (北京 399 信箱)

国外代号: 6210—MO