第32卷第4期 电子与信息学报 Vm．32NO．4

2010年4月 Journal of Electronics&Information Technology Apr．2010

基于可信计算的动态完整性度量架构

刘孜文∞ 冯登国@

(中国科学技术大学电子工程与信息科学系 合肥230027) (信息安全国家重点实验室中国科学院软件研究所北京100190)

摘要：该文提出一种基于可信计算的操作系统动态度量架构(DIMA)，帮助管理员动态地检查系统中进程和模块 的完整性。相对于以往的各种操作系统度量架构，该架构能按需对系统中活动的进程或模块进行动态实时的完整性 度量与监控，基本解决了其他架构难以避免的TOC-TOU问题，特别是针对某些直接对运行中的进程的攻击有很

好的效果。另外，DIMA实现了对对象细粒度度量——由度量整个文件实体细分为度量代码、参数、堆栈等等。

最后给出了基于Linux操作系统的动态度量原型实现，在实现中使用了基于可信平台模块(TPM)作为架构的信任 源点，测试结果表明DIMA能够实现预定目标且有良好的性能。 关键词：可信计算模块；完整性度量；动态完整性度量架构(DIMA)

中图分类号：TP309 文献标识码：A 文章编号：1009-5896(2010)04-0875-05

DOI：10．3724／SP．J．1146．2009．00408

TPM-Based Dynamic Integrity Measurement Architecture

Liu Zi．we产 Feng Deng-guo@

(Department of Electronic Engineering and Information Science，University of Science and Technology ol China，

Hefei 230027，China)

(State Key Laboratory of Information Security，Institute of Software，Chinese Academy of Sciences，

Beijino 100190，China)

Abstract：This paper presents a TPM-based architecture DIMA(Dynamic Integrity Measurement Architecture)， which helps the administrators check the integrity of the processes and modules dynamically．Compares with other measurement architectures，DIMA uses a new mechanism to provide dynamic measurement of the running processes and kernel modules．Some attacks to running processes which use to be invisible to other integrity

measurement architectures can be nOW detected．In this case．DIMA solves the TOC-TOU problem which always bothers others before．In addition，instead of measuring the whole file on the hard disk，the object is divided into some small pieces：code，parameter，stack and SO on to make a fine-grained measurement result．Finally，the DIMA implementation using Trust Computing Module(TPM)is discussed and the performance data is presented．

Key words：Trusted Computing Module(TPM)；Integrity measurement；Dynamic Integrity Measurement

Architecture(DIMA)

1 引言 如访问控制，远程证明等提供基本的安全数据。在 随着计算机应用的普及，对计算机系统的完整 可信计算的概念提出和推广后，度量作为可信计算

性保护日益受到重视，出现了各种完整性保护的模 的核心内容之一，其重要性得到了进一步提高。 型和实施方法。度量方法，是一个较新的方法。它 度量方法包括几个关键要素：度量的时间点，

的原理是在某些特定的时刻，对目标进行度量，得 度量的对象，度量在系统中实现的层次。其中度量

到目标的某些信息(比如对文件的散列值)，将这些 的时间点决定了在什么时刻进行度量；度量的对象

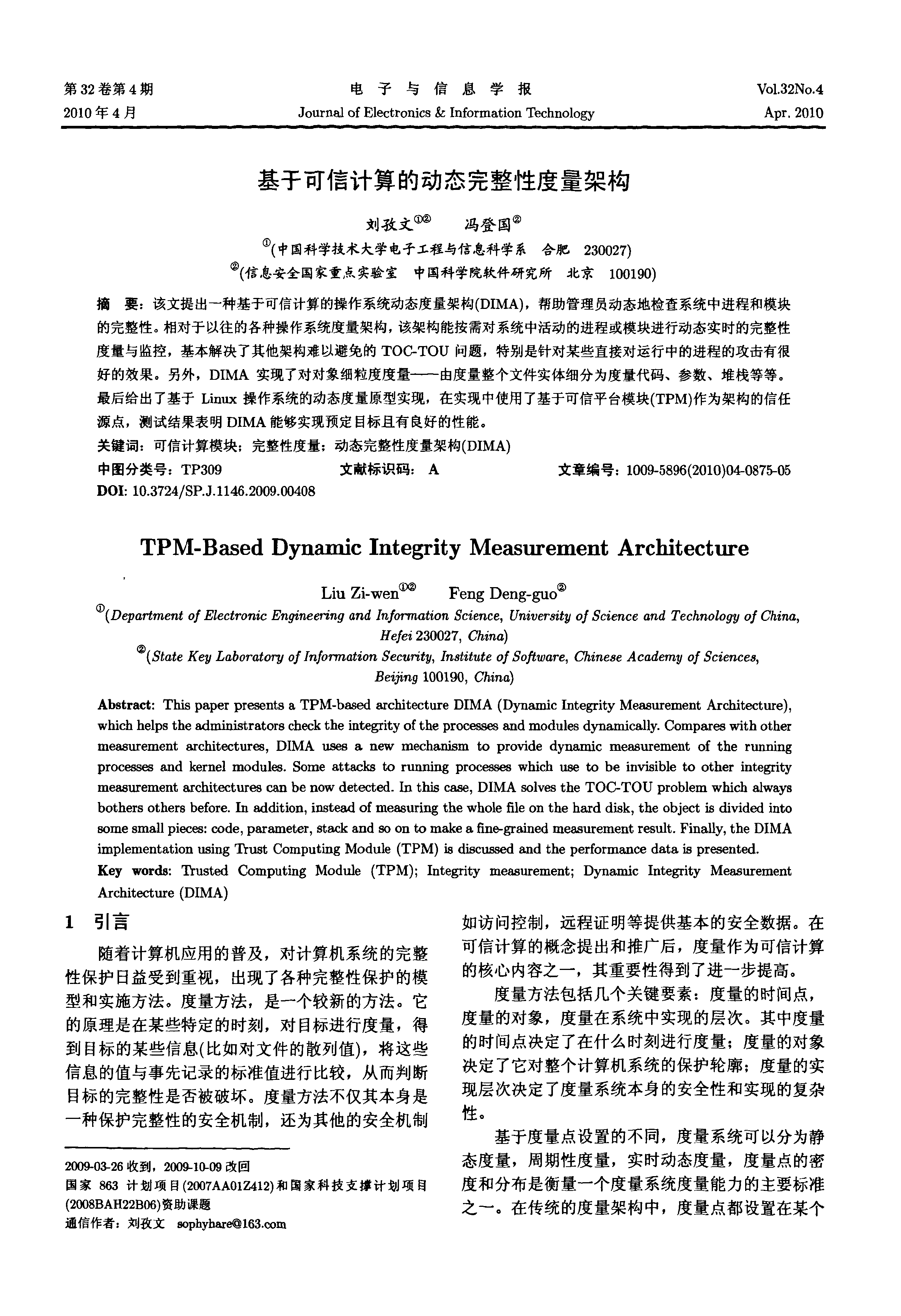
信息的值与事先记录的标准值进行比较，从而判断 决定了它对整个计算机系统的保护轮廓；度量的实

目标的完整性是否被破坏。度量方法不仅其本身是 现层次决定了度量系统本身的安全性和实现的复杂

一种保护完整性的安全机制，还为其他的安全机制 性。

基于度量点设置的不同，度量系统可以分为静

2009-03-26收到，2009-10-09改回 态度量，周期性度量，实时动态度量，度量点的密 国家863计划项目(2007AA012412)和国家科技支撑计划项目 度和分布是衡量一个度量系统度量能力的主要标准 f2008BAH22806)资助课题 之一。在传统的度量架构中，度量点都设置在某个 通信作者：刘孜文aophyhare@163．00m



万方数据

876 电子与信息学报 第32卷

或某些事先规定好的时刻(也就是静态度量或者周 量。IMA和PRIMA从本质上说，仍然是静态度量 期性的度量)。这种度量点的设置有如下几个不足： 系统。BIND[6】系统由程序员自己决定度量点并在度 f1)度量点由系统设计者决定，而不是由实际的使用 量点插入BIND提供的hook函数接口，从而提高了 者决定，因此在灵活性上受到影响：(2)遭遇典型的 度量精度。然而它增加了程序员编写程序的负担， TOU．TOC问题：进程在被度量前未受攻击，而在 且与以前的所有程序都不兼。LKIM[71以及后继的工 度量后遭到破坏，静态度量就会给出错误的结果， 作【81针对系统内核进行度量，它在静态度量的基础 周期性度量如果其度量周期长于进程完整性被损坏 上，定义一系列变量表示系统的状态，这些变量值 的时间，也可能发现不了这样的破坏。 发生变化的时候，重新进行度量，从而实现动态度

度量的另一个要素——系统中实现的层次的问 量的目的，它的主要不足如下：首先它强调的静态 题，在以前的大部分架构中都没有很好地得到解决， 度量加上状态变量不能表示真正的动态的度量；其

直到可信计算技术的出现。1999年TCPA(Trusted 次它度量针对的是Linux内核，对普通的进程无能 Computing Platform Alliance，后改名为TCG)[1l 为力。文献f91主要关注的是程序执行结果的远程证 制定了可信计算的标准，主要表述了如何利用可信 明，通过度量在整个执行期间目标进程与别的进程、 计算模块TPM[1j提供的各种功能来保护系统平台的 文件的数据交互，来对整个程序的执行过程进行证 安全。TPM本身提供了安全随机数产生器、非易失 明。 防篡改存储、密钥产生算法、RSA加解密密码函数 以上这些现有的各种度量系统在动态性和实时 以及hash函数SHA．1等功能，可以看出，TPM是 性方面都有很大的缺陷，而动态攻击并非不存在，

一种提供了某些安全功能的硬件(固件1，同时它的 这方面可以由自修改代码SMC技术中得到一些启 功能和规格比较统一，所以是设计各种计算机安全 示：自修改代码SMCllo](Self-Modifying Code)，是 系统的可信计算基TCB(Trust Computing Base)的 一类程序的统称，执行过程中会有意修改自身的代 很好选择。度量方法通过将度量系统的信任根构建 码，使得实际运行的代码和程序执行之前的代码的 在基于TPM的硬件环境中，能获得比以前各种基 静态二进制表示不相同。自修改代码可以有效隐藏 于纯软件的系统更好的安全性。 代码、数据以及程序的执行流程等，广泛应用于软 DigSig[21系统在可执行文件头部加入安全信息， 件保护和恶意代码等领域。SMC行为一般分为两 并在运行时验证这些安全信息，来解决对Linux可 步，首先修改存放代码的内存页的读写权限，然后 执行程序非法篡改的问题。缺点是DigSig自身的安 对代码进行覆盖，以达到自己修改代码的目的。一

全没有得到保障。CoPilot[3】系统独立于系统CPU之 种针对用户进程完整性的攻击也是基于类似原理， 外使用一个PCI板卡的协处理器进行度量工作，以 只是需要额外的步骤一一操作系统中为了保护进 确保度量系统本身的安全性。CoPilot的缺点在于设 程，一般都设定了不同进程的地址空间不能互相访 计实现复杂，需要进行多次对内存的映射操作，而 问，所以需要在修改读写权限之前，恶意进程需要 且只能先设定度量周期，不能真正实现动态的度量。 先冒充为目标进程。2．1节中说明了这种攻击的原

这两个系统代表了可信计算发展早期，度量方法中 理，4．1节实验将展示这种攻击。 纯软件的方法以及结合一些硬件的方法。第1种实 针对度量系统的实时性问题，本文设计了动态 现简单，但是度量系统以及度量结果本身的安全性 完整性度量架构(Dynamic Integrity Measurement

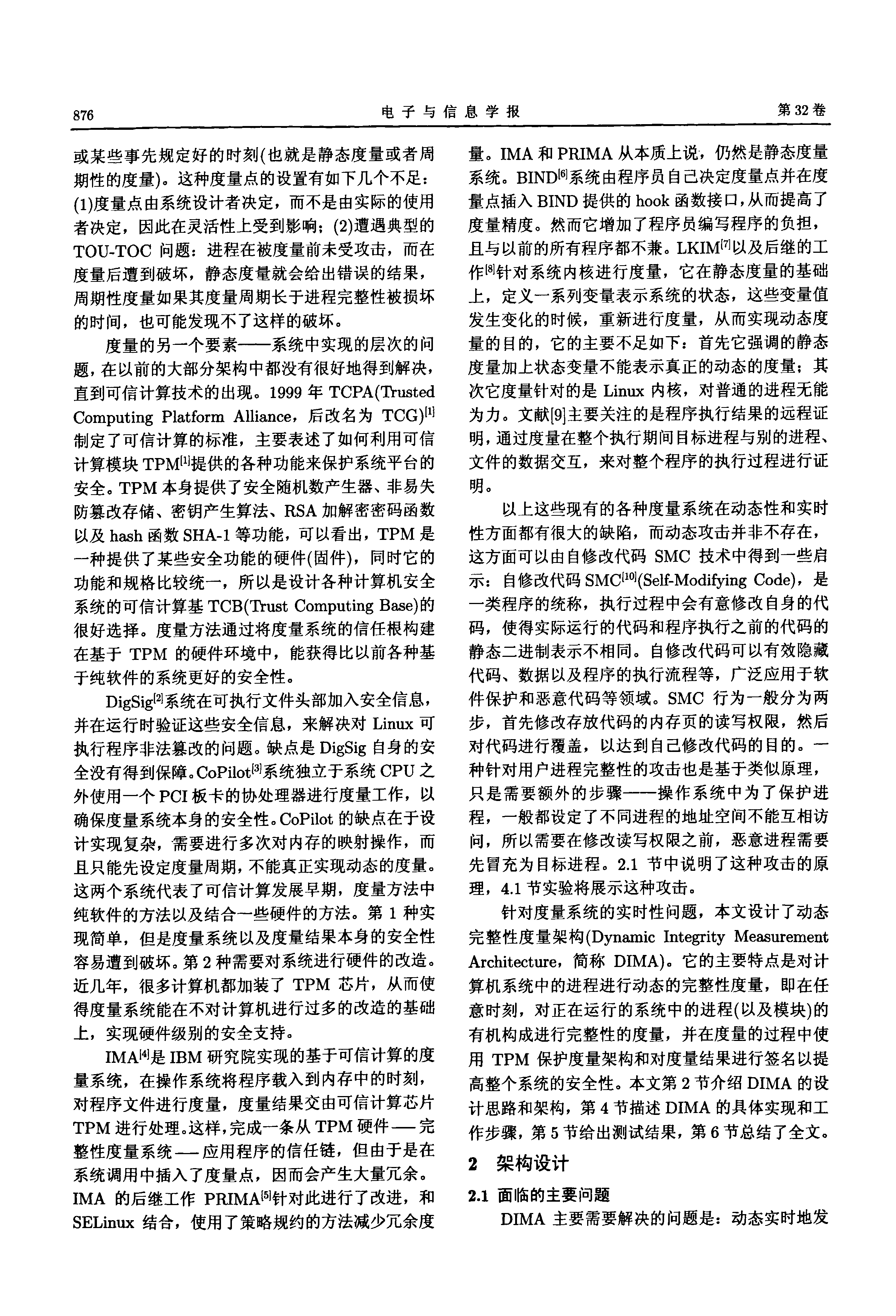
容易遭到破坏。第2种需要对系统进行硬件的改造。 Architecture，简称DIMA)。它的主要特点是对计 近几年，很多计算机都加装了TPM芯片，从而使 算机系统中的进程进行动态的完整性度量，即在任

得度量系统能在不对计算机进行过多的改造的基础 意时刻，对正在运行的系统中的进程(以及模块1的 上，实现硬件级别的安全支持。 有机构成进行完整性的度量，并在度量的过程中使 IMA[4】是IBM研究院实现的基于可信计算的度 用TPM保护度量架构和对度量结果进行签名以提 量系统，在操作系统将程序载入到内存中的时刻， 高整个系统的安全性。本文第2节介绍DIMA的设 对程序文件进行度量，度量结果交由可信计算芯片 计思路和架构，第4节描述DIMA的具体实现和工

TPM进行处理。这样，完成一条从TPM硬件一完 作步骤，第5节给出测试结果，第6节总结了全文。 整性度量系统一应用程序的信任链，但由于是在 2架构设计 系统调用中插入了度量点，因而会产生大量冗余。

IMA的后继工作PRIMA[SI针对此进行了改进，和 2．1面临的主要问题

SELinux结合，使用了策略规约的方法减少冗余度 DIMA主要需要解决的问题是：动态实时地发



万方数据

第4期 刘孜文等：基于可信计算的动态完整性度鼍架构 87'7

现和报告针对系统中已经运行的进程的完整性的攻 (3)关于度量架构的位置：事实上所有的安全功 击。例如，对于代码修改的方法而言：首先恶意进 能架构都面临自身安全的问题，因此有可信计算基 程伪装成目标进程欺骗操作系统，使得自己能够获 TCB(Trust Computing Base)的提出，只要TCB部 得对目标进程地址空间的内容进行修改的权限；接 分不被攻破，安全架构自身就是安全的。DIMA围 着对地址空间中存放代码的部分的页面属性进行修 绕可信计算芯片TPM来设计TCB，通过TPM对 改；然后对这些存放代码的页面内容进行修改，完 度量架构本身和度量结果做验证。 成恶意进程希望实现的功能；最后视情况将进程代 2．3架构 码恢复为原来未被修改的模样。整个过程见图1。 为了同时满足这些要求，遵循设计思路，DIMA

将跨度于用户空间和内核空间，整个架构见图2。

用户空间 内核空间

图1基于代码修改方法的攻击原理示意图

可以看出，这样的攻击有几个特点：一是入侵 的时机是进程运行之后，因此各种载入时的度量系 图2度量架构

统如IMA，PRIMA，DigSig等就不能检查出来，

对于周期检查的CoPilot系统，只要入侵完成在度 度量代理MA(Measurement Agent)处于用户

量周期之内，度量系统也是无能为力。二是入侵的 空间，用户的度量请求通过它的接口提交给内核，

对象是普通的用户进程，这样LKIM，BIND系统 后继工作中的远程证明等功能也是与它进行交互。 就不能有效的做出反应。三是入侵修改的内容是程 内核度量模块KMM(Kernel Measurement 序的代码部分，因为程序的代码部分一旦被修改， Module)，处于内核空间，完成整个动态度量的主体 整个程序的功能就会完全改变，从而实现入侵目的。 工作。可信启动以及信任链传递机制在事前的工作

2．2设计思路 中完成 ，作用是保证整个DIMA架构的安全，其

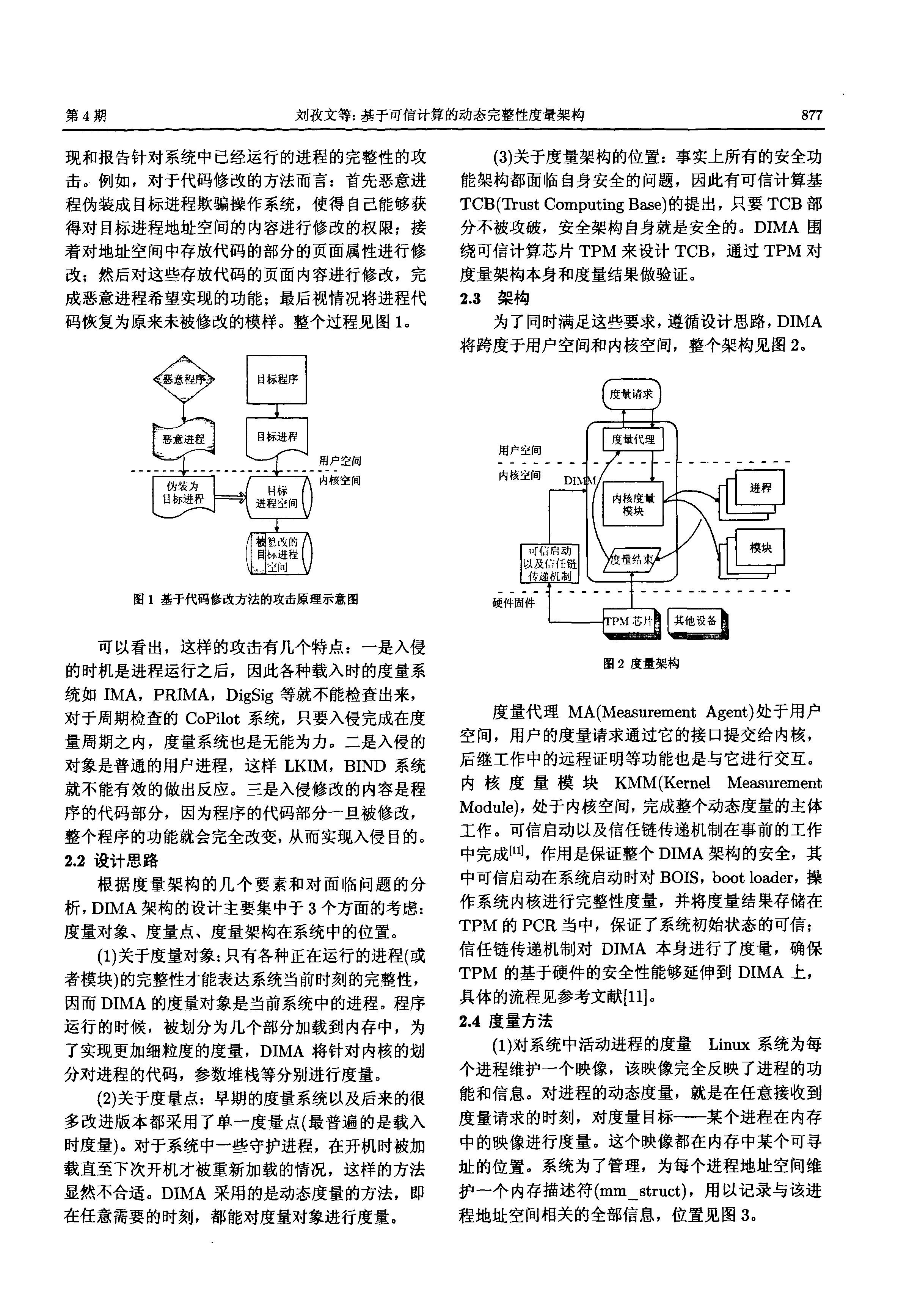
根据度量架构的几个要素和对面临问题的分 中可信启动在系统启动时对BOIS，boot loader，操

析，DIMA架构的设计主要集中于3个方面的考虑： 作系统内核进行完整性度量，并将度量结果存储在 度量对象、度量点、度量架构在系统中的位置。 TPM的PCR当中，保证了系统初始状态的可信；

(1)关于度量对象：只有各种正在运行的进程(或 信任链传递机制对DIMA本身进行了度量，确保 者模块)的完整性才能表达系统当前时刻的完整性， TPM的基于硬件的安全性能够延伸到DIMA上， 因而DIMA的度量对象是当前系统中的进程。程序 具体的流程见参考文献【11】。 运行的时候，被划分为几个部分加载到内存中，为 2．4度量方法 了实现更加细粒度的度量，DIMA将针对内核的划 (1)对系统中活动进程的度量Linux系统为每 分对进程的代码，参数堆栈等分别进行度量。 个进程维护一个映像，该映像完全反映了进程的功

(2)关于度量点：早期的度量系统以及后来的很 能和信息。对进程的动态度量，就是在任意接收到 多改进版本都采用了单一度量点(最普遍的是载入 度量请求的时刻，对度量目标——某个进程在内存 时度量)。对于系统中一些守护进程，在开机时被加 中的映像进行度量。这个映像都在内存中某个可寻 载直至下次开机才被重新加载的情况，这样的方法 址的位置。系统为了管理，为每个进程地址空间维 显然不合适。DIMA采用的是动态度量的方法，即 护一个内存描述符(ram struct)，用以记录与该进

在任意需要的时刻，都能对度量对象进行度量。 程地址空间相关的全部信息，位置见图3。



万方数据

878 电子与信息学报 第32卷

PAGE OFFSET

Env end (c)将它转换为物理地址并读取物理地址中内

NULL

环境字符串 env start 容，映射到KMM的进程空间中；

arg—end

命令行参数 (d)将这些内容交由TPM芯片做散列值，并由

arg\_start

拧序解释器的表 TPM进行签名，将散列值和签名值存入事先定义好

envp【1 ＆Ⅻ1vpl0】 的数据结构中。

argv[】 argy(0】 (3)对模块进行度量。接受到度量代理发送的度

argc

Start stack

返叫地址 量请求后经判定为模块，进行如下操作：

(a)解析请求中的模块名，并在内核维护的模块

栈顶单元

链表中找到模块，获得模块句柄；

图3进程地址空间 (b)通过句柄找到模块的正文的地址(\*module

—core)，长度core—text—size)

对于某一个进程而言，其运行过程和运行结果 (c)使用和度量进程一样的步骤进行度量。

可以进程地址空间内的数据决定。这些数据中，有 (4)度量架构的安全启动。操作系统启动之前，

几处数据最为关键。Start code和end code标识 由TPM对BIOS进行度量，启动BIOS后对Grub

程序代码所在的线性区的起始和中止地址；Start 进行度量，最后对操作系统的映像进行度量，从而

data和end data标识程序数据所在的线性区地址； 保证度量架构启动前的安全性，度量架构启动之后

Arg start和arg end标识命令行参数所在的堆栈 TPM对度量架构的关键部分如KMM模块和配置

部分地址。他们是度量的主体。 文件进行度量。具体的流程参见文献[n]。

(2)对内核模块的度量Linux系统为每一个模

4测试

块维护一个结构struct module，内核其他部分要与

模块进行数据交互可以通过访问代表这个模块的 4．1攻击测试 module结构进行；所有的模块都通过模块链表 使用基于SMC原理的攻击方法，对活动的进程 modules连在一起。DIMA通过对内核模块中这些 进行攻击，攻击进行中使用DIMA对此进程进行度 关键数据的度量，来实现对模块的度量。 量：在Linux下编译一个类似于hello world程序，

在打印“hello world”之前等待，直到控制台得到

3度量流程

键盘输入(模仿一个持续性的应用)。该程序等待时，

3．1度量代理MA(Measurement Agent) 使用基于SMC的方法修改代码段，这里只用全0

(1)格式化接收到的度量请求(可能来自于远端 覆盖，覆盖的长度按需要控制。使用DIMA对覆盖

程序的信任挑战，也可能来自于本地的安全需求)； 前和覆盖后的hello world程序进行度量。

(2)把请求发给内核的度量模块； 结果：无论长度如何，DIMA都能检测到入侵。 (3)待内核空间的KMM处理完后，把接受到的 覆盖的长度如果短，程序仍然正常运行结束，如果 度量信息组装，发送回远方或者本地。 覆盖的长度过长，则程序不能按照正常步骤输出

3．2内核模块KMM(Kernel Measure Module) hello world字样。可以看出，基于SMC的攻击动 态改变了程序的运行，但是DIMA能够很清楚的看

(1)通信。我们使用的是Linux提供的／proc虚

到度量值发生了变化。而在以往的架构中，做不到

拟文件系统机制：当KMM模块插入内核的时候， 这一点。作为对比使用了提供源码的IMA【4)架构进

在虚拟文件系统／proc中自动生成／proc／dynamic 行了测试，对该攻击，IMA不能检测到。

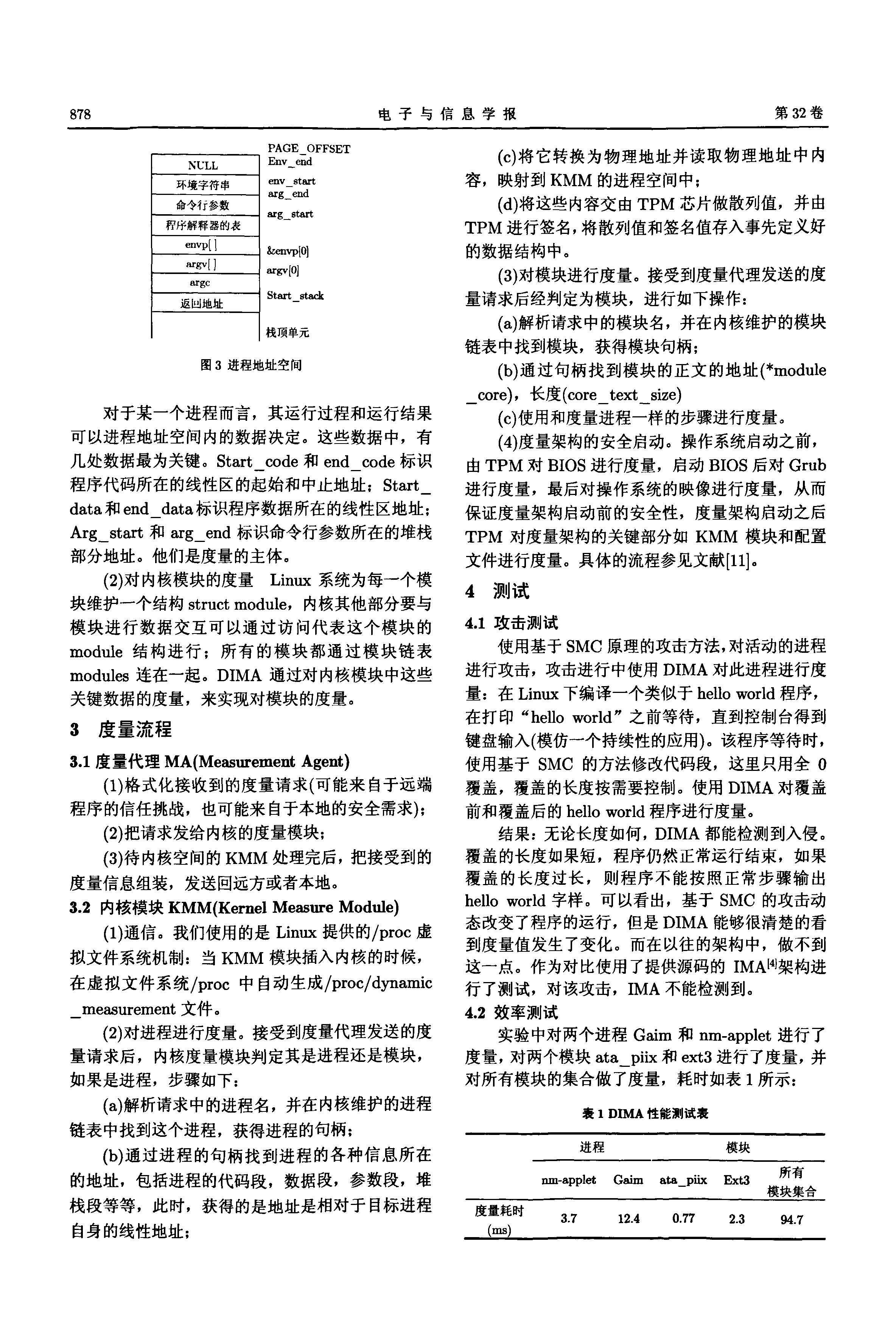
—measurement文件。 4．2效率测试 (2)对进程进行度量。接受到度量代理发送的度 实验中对两个进程Gaim和am—applet进行了

量请求后，内核度量模块判定其是进程还是模块， 度量，对两个模块ata piix和ext3进行了度量，并

如果是进程，步骤如下： 对所有模块的集合做了度量，耗时如表1所示： (a)解析请求中的进程名，并在内核维护的进程 表1 DIMA性能测试表

链表中找到这个进程，获得进程的句柄；

(b)通过进程的句柄找到进程的各种信息所在 的地址，包括进程的代码段，数据段，参数段，堆 栈段等等，此时，获得的是地址是相对于目标进程 自身的线性地址；



万方数据

第4期 刘孜文等：基于可信计算的动态完整性度量架构 879

由测试可以看出，对一个进程做度量所需要的 嘲 Jaeger T，Sailer R，and Shankar U．PRIMA：Policy-reduced

时间为10 ms数量级f视进程的大小而定)，对一个 integrity measurement architecture[C]．Proceedings of the 模块做度量所需要的时间为1 ITIS数量级，对系统造 eleventh ACM symposium on Access control model8 and 成的负担很小。 technologies．Lake Tahoe，California，USA，2006：19—28．

矧 Ski E，Perrig A，and Van Doom L．BIND：A fine-grained

5结束语

attestation service for secure distributed systems[C]．

本文介绍了一种基于可信计算的动态度量架 Proceeding of the IEEE Symposium on Security and Privacy． 构，并做了相应的实现。相对于现有的各种度量架 Oakland，CA，USA，IEEE Press，2005：154—168． 构，它最大的特点是实现了实时的动态度量，解决 研 Loscocco P A，Wilson P W，Pendergrass J A，and McDoneU

了度量中最关键的TOC—TOU问题，并把度量对象 C D．Linux kernel integrity measurement using contextual 扩展到进程，模块以及它们的周边信息，并利用可 inspection【c1．Proceedings of the 2007 ACM workshop on Scalable trusted computing．Alexandria，Virginia，USA，2007：

信技术芯片TPM进行硬件级别的保护。测试表明

2l一29．

DIMA能够完成对一些实时攻击方法的检测，性能

科 Thober M and Pendergrass J A．McDoneH C D：Improving

方面在对时间要求比较高的场合也有很好的表现。

coherency of runtime integrity measurement【C]．Conference

今后的工作包括几个方面，一是拓展对进程的度量，

on Computer and Communications Security Proceedings of

加入对进程其他安全信息的度量：二是考虑如何将

the 3rd ACM workshop on Scalable trusted computing．

此架构应用到访问控制之中，使得度量结果能得到 Alexandria，Virginia，USA，2008：51—60．

更为广泛的应用。 嘲 Gu Liang，Ding Xu-hua，Deng R H，Xie Bing，and Mei Hong． 参考文献 Remote attestation on program execution[C]．Conference on Computer and Communications Security Proceedings of the

Trusted Computing Group．TCG PC client specific

3rd ACM workshop on Scalable trusted computing．

implementation specification for conventional bios version 1．2，

Alexandria，Virginia，USA，2008：11—20．

July 2005． Wu

呷 Yong-dong，Zhao Zhi-gang，and Chui Tian-wei．An

Apvrille A，Gordon D，Hallyn S，Pourzandi M，and脚V．

attack on SMC-based software protection[M]．Springer Berlin

DigSig：Run-time authentication of binaries at kernel

／Heidelberg．2007：232—248．

Level[C]．Proceedings of LISA'04 Eighteenth Systems m 徐震，沈丽红，汪丹．一种可配置的可信引导系统．中国科学

Administration Conference． Atlanta， GA， USENIX 院研究生院学报，2008，25(5)：626—630．

Association November，2004：59-66． Xu Zhen，Shen Li-hong，and Wang Dan．LOIS grub：A Petroni N Jr and Fraser T．et 0L．Copilot——A coprocessor

confignrable trusted booting system[J]． Journal o，the

—based kernel runthne integrity monitor[C]．Proceedings of

Graduate School os the Chinese Academy oi Science，2008，

the 13th conference On USENIX Security Symposium．San

25(5)：626-630．

Diego，CA，2004，V01．13：13—13．

Sailer R，Zhang Xiao-lan，Jaeger T，and Van Doom L．Design

刘孜文： 男，1981年生，博士生，研究领域为操作系统安全、可

and implementation of a TCG-based integrity measurement

信计算．

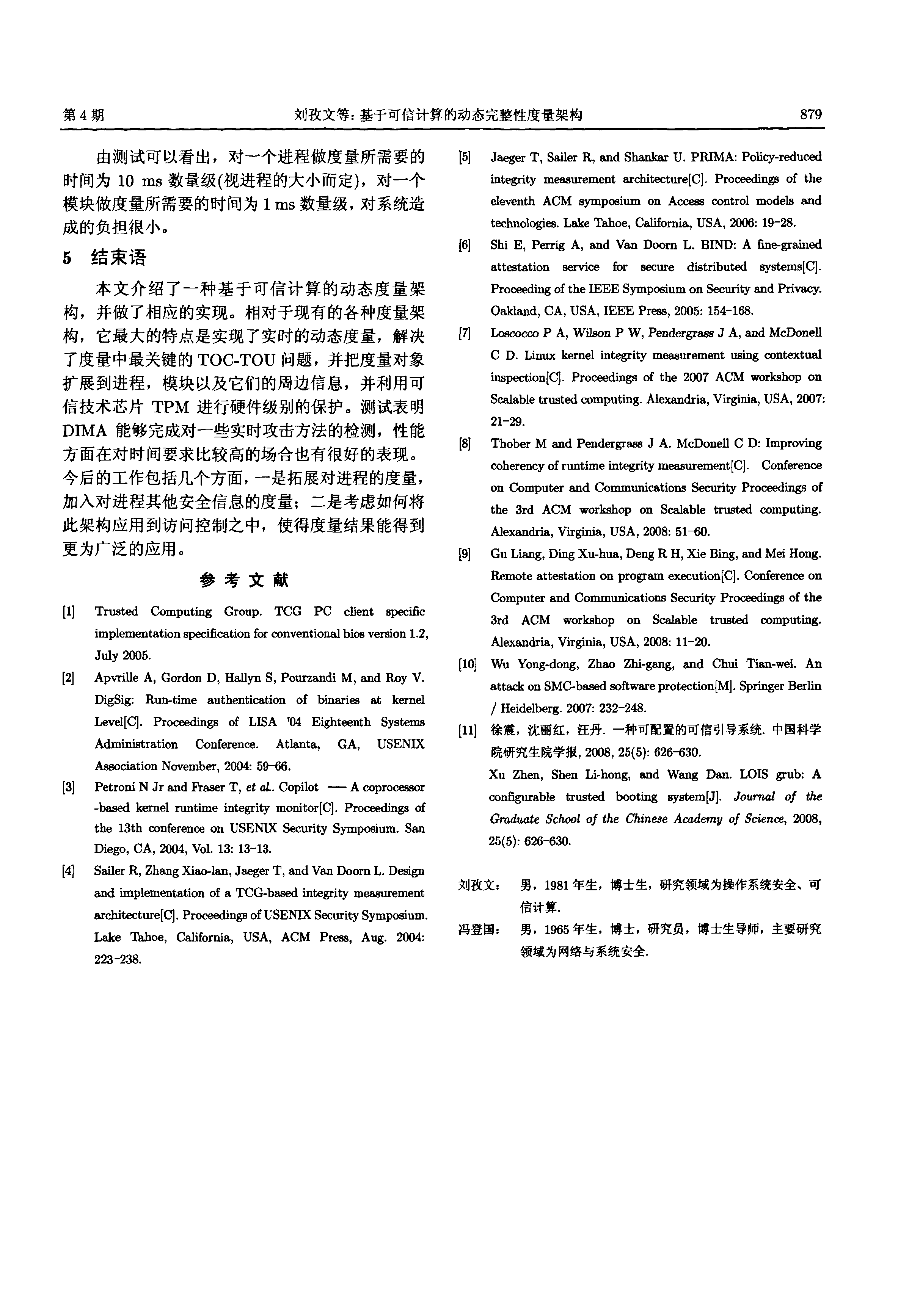
architecture[C]．Proceedings of USENIX Security Symposium．

冯登国： 男，1965年生，博士，研究员，博士生导师，主要研究

Lake Tahoe，California，USA，ACM Press，Aug．2004：

领域为网络与系统安全．

223-238．



万方数据