第 31 卷　第 6 期　世界科技研究与发展　Vol. 31　No. 6
2009 年 12 月　1069 - 1073 页　WORLD SCI-TECH R&D　Dec. 2009　pp. 1069 - 1073

远程证明方法的研究综述

陈　婷[1,2]　王永全[2]
（1. 华东理工大学信息科学与工程学院，上海 200237；2. 华东政法大学信息科学技术学院，上海 201620）

摘　要：远程证明方法是可信计算理论中一个重要的组成部分，本文追踪该领域的最新发展方向，重点剖析了四大类远程证明模型，并就其优缺点做出了详细的分析，横向上对所有模型进行了比较，从而进一步得到了现有远程证明模型的不足，展望了其未来的发展方向，为该领域的研究奠定了很好的基础。

关键词：可信计算；远程证明；TPM；校验和
中图分类号：TP309　　**文献标识码**：A

Survey of Remote Attestation

CHEN Ting[1,2]　WANG Yongquan[2]
（1. School of information science and engineering, East China University of Science and Technology, Shanghai 200237;
2. School of information science and technology, East China University of Political Science and Law, Shanghai 201620）

Abstract: One of the objectives of Trusted Computing is to provide remote attestation method that is able to confirm the status of remote platform or application. A number of typical remote attestation model and its related protocols are reviewed and compared in this paper, which conclude the existence of the problem in current model, and look forward to its future development direction.

Key words: trustworthy computing; remote attestation; TPM; checksum

1　引言

随着网络技术的不断发展，可参与网上信息交换的终端愈加丰富，在保证服务器提供的服务（或软件）以及网络安全的同时，也需要保证各种终端本身的可信性，因此，可信计算组 TCG 提出了可信计算远程证明技术[1,2]。远程证明被视为本地完整性验证的扩展，当代码发现被篡改，服务（软件）的执行就会失败。

2　远程证明基础

2.1　远程证明概念

一个验证实体能通过从不可信执行平台发来的证明确保软件的执行。向验证实体传输证明，保证远程不可信计算平台上的代码未被篡改，我们将这种行为称为远程证明。在远程证明过程中，待证明系统为了证明的需要，将系统所有模块信息和相关配置发送给质询方，向质询方提供其平台上自引导开始之后运行的所有组件信息（包括名字、版本等），质询方获得了待证明系统的相关信息，从而可以推断出系统的当前状态以及预测其后续行为。服务器利用远程证明机制限制客户应用，通过针对性地选择远程的应用程序，可以防止恶意程序或者可能被利用的有缺陷应用对服务的滥用，防止误用木马程序、避免与恶意的终端连接。通过这些限制，达到增强终端可信性，加强系统安全的目的。

2.2　篡改

当数据传输的内容被改变而未发觉，并导致一种非授权后果时便出现消息篡改。例如，消息“允许甲读机密文件”被改为“允许乙读机密文件”。目前防篡改保护主要是从硬件和软件方面来进行：

（1）硬件防篡改

可信计算组 TCG[1] 提出了一个重要的防篡改硬件，即可信平台模块 TPM，它是一块绑定在平台上的芯片。TPM 是 TCG 平台的主要组件，能产生随机数，提供加密功能（比如：SHA-1，HMAC，RSA 加解密，签名和密钥生成），以及防篡改的非易失存储器（主要用于密钥存储）。TPM 提供了一个平台配置寄存器集（PCR）用于存储平台配置的测量值。这些寄存器的内容仅能通过 extending 操作进行修改，或者通过重启机器将寄存器内容重置：PCRnew←SHA-1（PCRold||M），PCRold 表示寄存器原来的值，PCRnew 为寄存器的新值，M 是本次测量值，||代表将两个值连接。

在平台启动过程中，通过计算所有装载的平台组件的散列值，对平台的初始状态进行测量。作为启动时运行的第一段可信代码，可信测量根核（CRTM）测量 BIOS 的代码和参数，并将测量值记录在相应的 PCR 中。接下来，BIOS 测量引导器（bootloader）的二进制印象，同时也将测量值计入 PCR 中，之后将控制权交给引导器，引导器再对操作系统进行测量，由前一个程序度量后一个程序的完整性，只有在完整性通过验证后，才把控制权交给后一个程序，如此反复直到操作系统的启动。通过这种方法，建立一条从 CRTM 到操作系统和应用的信任链。

（2）软件防篡改

在许多可能的攻击中，认证的问题成为了焦点，比如，攻击的目的是篡改应用代码和数据，绕过认证许可，或者强制执行一个修改的未授权可执行程序。不同的软件防篡改方案用于保护软件。在文献[3,4]中详细描述了几种方法。混淆（obfucation）技术用于使应用代码变得晦涩难懂，从而增大了攻击者运用反向工程（reverse engineer）的方法搜寻代码具体含义的难度；混淆技术采用各种代码变形（transformation）的方法，在不改变代码的功能的前提下，改变源代码的结构。反向工程和除混乱化（de-obfuscation）的理论研究目前还处于早期阶段。二进制代码的混合、二进制数据的解析以及逆向编译在很多情况下是不可判定的。研究表明，去混淆化（de-obfuscation）技术是一个 NP 问题。

校验和方法(checksum)[5]是检验软件篡改领域的一个重要技术。这些方法将软件代码的一部分作为输入。如果输出的结果与预计算的结果不同,则说明软件被篡改。

2.3 远程证明协议所具有的实体

(1)质询方(challenger):验证平台信息(如平台配置、属性、软件的特征)的实体。

(2)证明平台(Attestation platform):测量平台的信息(如平台配置、属性、软件的特征),存储信息并向质询方证明信息的平台。

(3)可信第三方(Trusted Third Party,TTP):一个被质询方和证明平台认为可信的实体,可提供平台的身份证明,根据证明平台提供的配置信息提供相应的属性证书。

(4)属性证书(Attribute Certification):描述一个对象(平台或应用)的某个方面的行为,比如安全相关的需求,并以证书的形式保证该属性的真实性。

2.4 远程证明传输协议应该具有以下特性

(1)能证实协议通信双方主体的身份,确保身份不被冒充;

(2)保证证书的时效性,考虑到证书的撤销;

(3)确保信息在证明平台、质询方、可信第三方之间传输时不被篡改,即保证传输过程中信息的完整性;

(3)确保信息在证明平台、质询方、可信第三方之间传输时不被泄漏,即保证传输过程中信息的机密性;

(4)能够抵抗重放攻击,确保信息的新鲜性;

3 远程证明典型模型

3.1 基于二进制的远程证明模型

基于二进制的远程证明模型是应用 TCG 远程证明规范来对所有用户组件进行证明。在终端可信链的建立过程中,任何将要转移控制权的实体(可信链的前一环),在控制权转移到下一个实体前(可信链上的后一环)都必须对该实体进行可信度量,如果符合某种要求,比如当前度量值和一个事先保存的预期度量值一致,则控制权才能发生转移,如创建用户级别进程时,内核测量所有装载进入该进程的可执行代码(如原始可执行共享库),该可执行代码随之可测量它装载的敏感输入(如参数,配置文件,shell 过程等)。所有的测量结果存储在指定 PCR 寄存器和 SML 中,在第 1 节中对 PCR 寄存器进行了详细地阐述。有学者也称之为装载时证明(load - time attestation),因为其归根到底是对二进制代码进行测量,并用该测量值对平台配置进行验证。对 TCG 的远程证明规范,其基本步骤如下[6]:

(1)A 产生一个不可预测的 160bit 随机数 nonce;

(2)A 将随机数 nonce 传送给 B;

(3)B 中的 TPM 产生验证身份密钥对{VK},并向 CA 获取 AIKpub 的证书 cert(AIKpub);

(4)B 用 AIKpriv 对 PCR 值和随机数签名,Quote = sig{PCR,nonce} AIKpriv;

(5)B 获得 SML 测量日志;

(6)B 将得到的数字签名,SML 日志和 AIK 公钥证书传送给 A;

(7)A 验证 AIK 公钥证书的真实性;

(8)A 验证数字签名的可信性;

(9)A 验证随机数值是否正确,从而验证该会话的新鲜度,且将 SML 与 PCR 值对比,如果相等,则说明该平台未被篡改。

基于 TCG 的远程证明模型的优点:运用了防篡改硬件 TPM,进行系统的测量、存储和报告,保证了数据的完整性和机密性;提供了基于身份的数字签名,确认了平台的身份;

基于 TCG 的远程证明模型存在着一些弊端。第一,每一个组件的新版本拥有不同的二进制代码,从而产生不同的散列值,导致存在大量的配置。第二,因为平台配置的测量是在系统启动时进行的,这种证明模式只能提供静态的证明,不能保证运行时完整性。第三,证明的过程需要提供 PCR 值,也即平台配置状态,从而暴露了平台的配置(比如平台上运行的软件版本等等),容易受到攻击。第四,远程证明的目的是要证明某软件是否执行在某个指定状态,内容提供商很容易利用这个功能,指定相应的软件服务商,如果平台运行的软件非内容提供商所指定,则不予提供服务,这容易造成软件服务商与内容提供商勾结,从而造成垄断。第五,该模型需要硬件,即 TPM 的支持,不具备向前的兼容性。第六,模型的远程证明协议容易遭受重放攻击。

3.2 混合远程证明模型

为了克服基于 TCG 远程证明存在的缺点,大量的混合远程证明模型被提出,这些模型中应用到了基于 TCG 远程证明模型的特点,运用 TPM 安全存储代码散列值,并进行了一些相应的扩展,防止 AP 的配置被 AR(Attestation Requester)或者其他攻击者获得。

语义远程证明模型[7]将远程证明与虚拟技术相结合,运用基于语言的虚拟机技术,提出了复杂、具有动态性、高级别程序属性、平台无关性的远程证明模型。通过在 AP 建立 JAVA 虚拟机——TrustedVM,并在虚拟机中执行平台无关的 JAVA 代码,测试该 AP 的程序的行为、类属性、动态属性、系统属性等,并将测试结果返回给 AR。AR 根据其安全策略,判断该 AP 是否可信。语义远程证明模型可应用于多种不同应用中,该模型在 P2P 协议中的执行步骤如下:

(1)服务器 B 将协议观测点(protocol watcher)的 Java 字节代码发送给 A,A 上运行有 TrustedVM:$B \xrightarrow{\text{protocol watcher}} A[\text{TrustedVM}]$;

(2)A 向 B 发送服务请求:$A[\text{TrustedVM}] \xrightarrow{\text{server request}} B$;

(3)协议观测点观测到该请求,检查该协议消息是否与安全策略一致,并报告结果。如协议观测点检查所有客户端发出的消息,并检查其两个属性:

1)pong 消息,确保消息中的 IP 地址与 A 一致

2)query hit 消息,检验查询的文件在 A 中确实存在

语义远程证明模型的优点在于:使用 TrustedVM 测试证明平台行为的可信性,将平台的行为与身份分离;运用平台无关的 Java 虚拟机,使该模型能应用于各种系统环境中,它能与原有操作系统进行很好的结合。可测试证明平台的各

种动态属性。实验数据表明,该模型的执行对系统的性能影响较小。

语义远程证明模型的缺点:系统只对高级别的应用做证明,而从系统引导(boot)至操作系统的可信性,仍然只按照 TCG 的远程证明标准进行,因此,也存在 TCG 远程证明的缺点;协议观测点并不是对所有消息进行观测,如多跳消息(multi-hop messages);没有将验证平台的行为信息与身份信息绑定,证明模型没有体现平台身份。

另一类混合型的远程证明模型为基于属性的远程证明模型[8,9],针对 TCG 远程证明模型中直接将系统配置散列值直接传递给验证方验证的缺点,将系统配置传递给可信第三方(TTP),由 TTP 得出其具有的属性,并发放相应的属性证书,从而在验证方需要验证平台可信性时,将属性证书发送给验证方以证明平台的可信性。

其模型结构如图 1 所示。

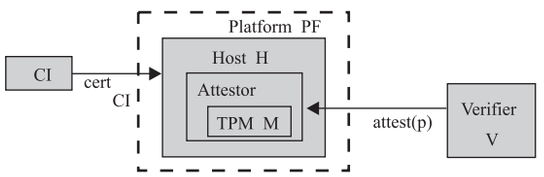

图 1　基于属性的远程证明模型

Fig. 1　Abstract model of property-based remote attestation scenario

模型定义数字签名模式为三元组:(GenKey(),Sign(),Verify()),他们依次代表密钥生成、签名和验证。Sign(K-1,m,σ)表示使用私钥 K-1 对信息 m 签名,输出 σ。ind ← Verify(K,m,σ)表示使用公钥 K 对签名验证,函数的返回值 ind ∈ (True,false)表示验证的结果为真或假。其证明协议中运用 Pederson 的承诺模式[10],对系统配置值进行承诺,并通过 CL 签名[11]对 TTP 的属性证书进行信息隐藏,用零知识证明协议作为证实方(attestor)和验证方(verifier)之间的信息交互协议。

(1)证书发布者 CI 选取 $R_0, R_1, S, Z \in QRn$,产生密钥对$\{vk_{CI}, sk_{CI}\}$,公钥 vkCI = (n, R_0, R_1, S, Z),密钥 $sk_{CI} = p$;

(2)验证平台的 TPM 将平台的 PCR 值,用 cs 表示,发送给 CI,CI 根据 cs 得到其具有的相应属性 ps,颁发属性证书 $\sigma_{CI} \leftarrow IssueCertCI(sk_{CI}, (cs_i, ps))$;

(3)CI 任意选取素数 ei,随机数 vi,计算 Ai,使其满足 ,σ_{CI}用 Ai,ei,vi 表述为,$\sigma_{CI} = (Ai, ei, vi)$;

(4)质询方将随机数 nonce 传送给验证平台;

(5)验证平台向 CI 取得属性证书 σ_{CI};

(6)验证平台将属性证书 σ_{CI}用信息隐藏的方式进行转换,取随机数 w,计算 $\hat{A} = A_i S^w \bmod n, \hat{v} = v_i - we_i$,从而得到隐藏的签名证书 $\hat{\sigma}_{CI} = (\hat{A}, e_i, \hat{v})$;

(7)验证平台的 TPM 用 Pederson 承诺模式,对平台的配置 cs 进行转换,保证得到的结果确实属于验证平台,但不暴露其真实的值。并用 TPM 的身份验证密钥对该结果签名,连同验证平台将属性证书以及 nonce 一同发送给质询方;

(8)质询方验证平台的身份,并确认发送的配置值确实来自指定验证平台,且真实有效;

(9)质询方验证属性证书的有效性;

(10)质询方检查得到的 nonce 值是否与发送给验证平台的相等,如果相等,则证明该会话的新鲜度。

基于属性的远程证明模型的优点:向验证方隐藏了系统平台配置,从而避免了系统配置的泄漏;验证方只通过属性证书验证系统平台的安全性,不需要了解系统复杂的配置,如系统软件运行的版本号,减少了软件行业垄断现象的发生;系统平台在打补丁、升级等状态下时,可灵活的处理失效的系统配置,其相应的属性证书的撤销不需要经过 TTP。

基于属性的远程证明模型的缺点:需要借助 TTP 来产生属性证书,从而将 TTP 纳入 TCB(可信计算基)中,加大了 TCB 的复杂性;需根据系统配置动态判断其相应属性,没有成熟的理论来实现;该模型只能验证系统的静态信息,即平台的配置等信息,无法体现系统的动态行为。

3.3　基于软件的远程证明模型

Seshadri 等人在不需要 TPM 的前提条件下,提出了一个嵌入式设备的远程证明方案[12]。之后,他们又提出了结合遗留系统的远程代码完整性验证解决方案,Pioneer[13],其中包含两个阶段的 challenge-response 协议。首先,验证者(verifier)确信不可信(untrusted)主机上有一个验证代理运行;然后,验证代理向验证者报告不可信主机上可执行代码的完整性。详细的步骤如下(如图 2 所示):

(1)验证者向不可信主机发送质询 n,请求执行一个验证代理 V,并记录其开始执行时间:$t_1 \leftarrow$ tcurrent。

(2)以质询 n 为种子,采用伪随机路径遍历验证代理的内存。在该路径基础上,计算校验和(checksum):c ←cksum(n,V)。

(3)验证代理向验证者报告校验和 c。验证者检测验证代理是否满足以下两个条件,从而判断其完整性:(a)验证代理的签名传递的时间为 $t_2 \leftarrow$ tcurrent,验证者知道校验和计算执行时间上限,时间需满足:$t_2 - t_1 < \Delta t_{expected} = \Delta t_{cksum} + \Delta t_{network} + \delta t$,$\Delta t_{cksum}$是预期的校验和功能执行时间,$\Delta t_{network}$是网络延时,δt 是一定的时间误差;(b)验证者用本机的验证代理进行相同的校验和计算,计算的结果须与不可信主机上的校验和计算结果一致。

(4)不可信主机上的验证代理对可执行代码 E 计算其散列值,h ← hash(n,E)。

(5)散列值 h 被发送给验证者,后者进行验证。同样,验证者通过本地的验证代理对可执行代码 E 进行同样的散列计算,将计算结果与 h 比较,如果一致,则代表该应用 E 是可信的。

(6)在不可信主机端,验证代理调用应用 E,并将控制权递交给 E。

该模型用到了校验和方法(chsum())[5],在软件防篡改领域,校验和方法应用较多,该方法读取软件代码,如果其计算的结果与预先估算的值不一致,则说明该软件已被篡改。该方法存在很多的限制:

(1)校验和代码的执行需要时间优化。如果一个攻击者能优化校验和代码,他将获得进行恶意行为的时间。

(2)为了最大化攻击者的工作量,校验和函数需采用伪随机遍历的方式读取内存,以阻止了攻击者预测内存读取的

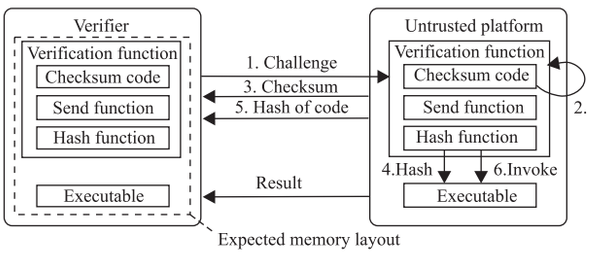

图 2　Pioneer 远程证明模型

Fig. 2　Overview of Pioneer

行为。在 Pioneer 中,质询 n 是伪随机遍历的种子。

(3)校验和代码的执行时间必须能预测。因此,Pioneer 需要以超级模式运行,且不能被打断。

Pioneer 模型的优点在于:不需要硬件的支持,以基于软件的方法实现了远程证明,考虑到了时间的因素,采用校验和方法,用伪随机遍历的方法检测内存,从一定程度上减少了攻击者通过内存复制攻击达到软件篡改的目的。而这种方法不需要涉及可信第三方,模型相对较简单。

其主要缺点是:模型中加入了网络延时 $\Delta t_{network}$,而网络延时因为各种未知的原因,很难预测,因此,难以实际运行;其次,校验和检测方法在实践上存在着限制;第三,验证代理本身的可信性未被验证。第四,模型中没有涉及到身份的证明,很容易遭到冒充攻击。

3.4　其他的远程证明模型

最近,有学者提出了基于行为的远程证明模型[2,14]。TCG 用实体行为的预期性来定义可信:一个实体是可信的,如果它的行为总是以预期的方式达到预期的目标。这一定义的优点是抓住了实体的行为特征。基于行为的远程证明模型依据 TCG 的这一定义,将平台状态证明转化为对平台历史行为序列的可信证明,有效地避免了在准确描述计算平台状态方面的难题,保证了模型实现的可行性和可扩展性,并且不会暴露证明平台的配置信息。

该模型通过一个行为度量模块收集系统的行为信息,将对证明平台的可信判别转化为对证明平台的系统行为历史进行可信度量验证。

定义 1 计算平台中一次特定系统行为 a 是可信的是指行为 a 符合质询方的可信行为预期策略。

定义 2 如果平台状态 ri 可信,系统行为 a 是可信的,并且 $T(a, r_i) = r_{i+1}$,那么平台状态 r_{i+1} 可信。

定理 1(计算平台状态可信判别定理)如果平台在某一时刻 t 的状态 r_t 可信,并且 t 时刻之后平台系统行为序列 $A = a_1 a_2 \cdots a_n$ 中的每一个系统行为 ai($1 \leqslant i \leqslant n$)都可信,那么系统行为序列 A 发生后的平台状态 $r_{t+1} = T(A, r_t)$ 是可信的。

定义 3 假设 U 表示与平台状态可信相关的系统行为集合,A1 和 A2 是两个系统行为序列,如果下列条件同时满足,称 A2 是 A1 的可信相关行为序列:

(1)对于 A1 中的任何一个系统行为 a,如果 $a \in U$,那么 a 必然在 A2 中;

(2)对于 A1 中的任何一个系统行为 b,如果 $b \notin U$,那么 b 必然不在 A2 中;

(3)A2 中各个系统行为的顺序与它们在 A1 中的顺序一致。

定理 2 假设 As 是系统行为序列 A 的可信相关行为序列,如果平台在某一时刻 t 的状态 r_t 可信,并且组成 As 的每一个系统行为都可信,那么系统行为序列 A 发生后的平台状态 $r_{t+1} = T(A, r_t)$ 是可信的。

基于行为的远程证明模型的优点:首先,对证明平台上用户的故意安全策略冲突行为进行管理和控制,很好地保护用户平台配置的隐私性和平台配置的灵活性;其次,由于它只对与证明平台可信状态相关的系统行为进行分析,所以它能够提高证明要素的精度和效率,并减少证明过程中所需的计算量;第三,为该模型的可执行性提供了可靠的理论证明。

基于行为的远程证明模型的缺点:对系统行为记录,增加了系统的开销,影响了系统的性能;其可信行为预期策略采用白名单的形式,要求对所有行为都已知,事实上很多行为都是未知的,因此,在实现上存在着困难;该模型只考虑到系统的动态行为的可信性,而未考虑到系统的静态可信性,如启动状态的可信性,内核的可信性等,没有从发生行为的基础上作可信性判断。

4　远程证明模型存在的问题

远程证明方法是可信计算理论中一个重要的组成部分,本文追踪该领域的最新发展方向,重点剖析了四大类远程证明模型,并就其优缺点做出了详细的分析,横向上对所有模型进行了比较,从而进一步得到了现有远程证明模型的不足:

(1)缺少统一的建模

现有的研究从不同的侧重点对该领域进行了研究,分别从二进制代码本身、代码属性、嵌入程序运行迹等方法进行远程证明建模,但没有一个通用的证明模式,且各种方案都存在着缺陷。每种模型都缺少有力的模型分析以证明其正确性、有效性、安全性。

(2)不能兼容于遗留操作系统

以上方案中的绝大多数方案都面临着不能在遗留系统中运行的事实,使其只能停留在理论层次上,无法实际应用。而 Pioneer 项目虽然能够结合,却因为其需要众多的假设前提,而无法实现。所有这些解决方案需要证明服务运行在一个安全的可执行环境中,它们不容易在遗留的操作系统中执行。

(3)缺少对系统的动态测量

大多数远程证明模型只能对系统的静态特征作验证,如系统启动时状态,而无法体现其动态特征,因此无法真正的对系统的所有状态作全面验证。

(4)缺少实际应用

众机构注重于理论上的研究,未能实现其支撑的应用平台,并结合实际的应用。

5　展望

结合第 3 节中的问题,我们对远程证明建模可以从下面几个方面作进一步的工作:

(1)进一步研究远程证明模型所需具备的条件,尤其是动态测量、存储以及报告机制的合理性,这是远程证明所必需的基础。

(2)通用证明模型的研究。结合各种模型的优点,综合模型实际需要,将适合于某一特定需要的证明模型扩展到更广泛的应用中,如数字版权管理 DRM 中。

(3)使模型的实现更具实际意义,不仅能应用到具有特定硬件(如 TPM)的设备中,且可以在各种系统中使用,更能结合原有的遗留系统,达到证明的目的。

(4)模型的评价问题。如何对众多的模型进行客观的性能评测也是一个值得研究的方向,目前各种模型给出的性能评测数据都是以实验室测试数据为主。

(5)结合其它学科的知识,如虚拟技术,软件行为学,机器学习等,继续探索适合远程证明的新模型。

6 结束语

证明模型的研究是当今信息安全领域的热点,通过对现有的几类证明模型进行比较分析,我们发现其各有所长,也存在着许多不足,从而更明确了远程证明领域研究的几个重要的方向。结合虚拟技术,进行通用模型的研究,是我们下一步的研究工作。

参考文献

[1] Trusted Computing Group. TCG Architecture Overview[EB/OL]. http://www.trustedcomputinggroup.org,2004 - 04

[2] Zhang Huanguo, Wang fan. A behavior-based remote trust attestation model[J]. Wuhan University Journal of Nature Sciences, 2006, 11(6):1819 ~ 1822

[3] Collberg C, Thomborson C. Watermarking, tamper-proofing, and obfuscation-tools for software protection[J]. IEEE Transactions on Software Engineering, 2002, 28(8):735 ~ 746

[4] Naumovich G, Memon N. Preventing piracy, reverse engineering, and tampering[J]. IEEE Computer, 2003, 36(7):64 ~ 71

[5] David Aucsmith. Tamper Resistant Software: An Implementation[A]. In: R J Anderson, editor, Proceedings of First International Information Hiding Workshop (IHW)[C]. Cambridge, England, 1996, published in Lecture Notes in Computer Science (LNCS), 1997, 1174:317 ~ 333

[6] R Sailer, X Zhang, T Jaeger, L van Doorn. Design and implementation of a TCG-based integrity measurement architecture[A]. In: Proceedings of the 13th USENIX Security Symposium[C]. USENIX, Aug, 2004, 223 ~ 238

[7] Vivek Haldar, Deepak Chandra, Michael Franz. Semantic Remote Attestation - Virtual Machine Directed Approach to Trusted Computing[A]. In: Proceedings of the 3rd Virtual Machine Research and Technology Symposium[C]. San Jose, CA, USA, 2004, 29 ~ 41

[8] Sadeghi A, Stüble C. Property-based attestation for computing platforms: caring about properties, not mechanisms[A]. In: C. Hempelmann and V. Raskin, editors, Proceedings of the new security paradigms workshop 2004[C]. Nova Scotia, Canada, 2004, 67 ~ 77

[9] Liqun Chen. A Protocol for property based attestation[A]. In: Proceedings of the 1st ACM Workshop on Scalable Trusted Computing (STC)[C]. Alexandria, Virginia, USA. ACM. Nova Scotia Canada, 2006, 7 ~ 16

[10] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing[A]. In: J Feigenbaum, editor, Advances in Cryptology - CRYPTO '91[C]. volume 576 of LNCS, pages 129 ~ 140. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1992. Extended abstract

[11] Camenisch J, Lysyanskaya A. A signature scheme with efficient protocols[A]. In: Third Conference on Security in Communication Networks - SCN '02[C]. volume 2576 of LNCS, pages 268 ~ 289. Springer-Verlag, Berlin Germany, 2002

[12] Seshadri A, Perrig A, L van Doorn, P K Khosla. SWATT: SoftWare-based Attestation for Embedded Devices[A]. In: 2004 IEEE Symposium on Security and Privacy (S&P 2004)[C], Berkeley, CA, USA, 2004, 272 ~ 282

[13] Seshadri A, Luk M, Perrig A, L van Doorn, P. Khosla. Externally verifiable code execution[J]. Communications of the ACM, 2006, 49(9):45 ~ 49

[14] 李晓勇,左晓栋,沈昌祥. 基于系统行为的计算平台可信证明[J]. 电子学报,2007,35(7):1234 ~ 1239

作者简介

陈婷(CHEN Ting,1979-),女,工程师,博士研究生,主要研究方向为信息安全,可信计算。

王永全(WANG Yongquan,1964-),男,教授,硕士生导师,主要研究方向为网络与信息安全(计算机取证)、模糊集及粗糙集理论与人工智能、理论计算机科学(序结构、逻辑代数)等。

(责任编辑:高利丹)