doi:10.11959/j.issn.1000-436x.2015139

基于 TCM 的安全 Windows 平台设计与实现

冯伟，秦宇，冯登国，杨波，张英骏
（中国科学院软件研究所 可信计算与信息保障实验室，北京 100190）

摘 要：为了解决 Windows 系统的完整性度量与证明问题，提出了一种基于可信密码模块 TCM (trusted cryptography module)的安全 Windows 平台方案。通过扩展 Windows 内核实现了 2 种安全模式：在度量模式下，所有加载的可执行程序都会被度量，度量值由 TCM 提供保护和对外认证；在管控模式下，度量值会进一步与管理员定制的白名单进行匹配，禁止所有不在白名单中的程序执行。实验分析表明，该方案可以增强 Windows 系统的安全性，抵抗一些软件攻击行为；同时，系统平均性能消耗在 20~30 ms 之间，不会影响 Windows 的正常运行。

关键词：可信计算；完整性度量；可信密码模块；Windows 安全
中图分类号：TP309 **文献标识码**：A

Design and implementation of secure Windows platform based on TCM

FENG Wei, QIN Yu, FENG Deng-guo, YANG Bo, ZHANG Ying-jun
(Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Science, Beijing 100190, China)

Abstract: A secure Windows platform solution based on TCM was proposed to solve the integrity measurement and attestation problem of the Windows system. Two security modes were realized by extending the Windows kernel: in the measurement mode, all executable contents that were loaded onto the Windows system were measured, and the TCM provided the protection and outward attestation for these measurements; and in the control mode, the measurements were further compared with a whitelist customized by an administrator, and all the programs that were not included in the whitelist would be prohibited from running. Experiment analysis shows that proposed solution can enhance the security of Windows platform and resist some software attacks; and at the same time, the average performance overhead is about 20~30 ms, which will not influence the normal running of Windows.

Key words: trusted computing; integrity measurement; trusted cryptography module; Windows security

1 引言

无论对于个人还是企业，Windows 都是应用最广泛的操作系统。特别是一些传统行业和政府企业部门，都使用比较旧的 Windows 版本，如 Windows XP、Windows Server 2003 等。根据北信源安全公司发布的调研报告[1]，政府企业中 Windows XP 系统的使用比例甚至高达 72.6%，这些系统通常承载着一些核心业务和安全服务。例如，目前很多 ATM (automatic teller machine)机还是基于 Windows XP 系统。这些终端系统经常面临各种攻击，如恶意代码入侵、伪造非法的终端设备、数据和系统的非法篡改与破坏等。特别地，随着微软宣布停止对 Windows XP 的支持和更新服务，这些终端系统面临更大的安全威胁。

可信计算[2~4]是一种有效的安全技术，能为终

收稿日期：2014-06-25；修回日期：2014-08-12
基金项目：国家自然科学基金资助项目（61202414, 91118006）；国家重点基础研究发展计划（“973”计划）基金资助项目（2013CB338003）
Foundation Items: The National Natural Science Foundation of China (61202414, 91118006); The National Basic Research Program of China (973 Program) (2013CB338003)

端系统环境提供可信性保证，并能将这种可信延续到网络。具体地，可信计算以防篡改的硬件芯片为信任根，通过可信度量技术收集终端平台的软硬件状态，并能使用密码算法和协议向远程方提供终端软硬件状态的证据。目前存在的可信度量技术[5,6]主要以 TPM (trusted platform module)作为信任根，针对 Linux 系统进行设计和实现，只对终端系统环境进行度量，而缺少根据度量结果对终端系统的管控功能。另外，终端应用程序的复杂性和多变性也是影响可信度量技术使用的一个障碍。为此，本文针对一些特殊的 Windows 应用场景(如 ATM 服务终端，其应用软件稳定、不常更新)，以我国可信密码模块 TCM (trusted cryptography module)作为信任根，设计并实现了 Windows 操作系统下的可信度量机制，具体贡献如下：

1) 本文的度量机制能让政府企业的一些关键服务（如 ATM 系统）继续使用 Windows XP 操作系统，同时为 Windows 操作系统提供基于可信度量技术的安全防护，能抵抗恶意代码、非法的终端设备、数据和系统的非法篡改与破坏等攻击行为；

2) 以 TCM 安全芯片作为信任根，通过扩展 Windows 内核模块对系统可执行程序进行完整性度量，并基于白名单机制设计了终端的自动管控功能；

3) 首个完整地实现了 Windows 操作系统的可信度量技术，并对原型系统进行了详细的评估，包括安全性分析、性能评估和攻击实验等。

2 背景和相关工作

2.1 可信计算

可信计算[7]通过在计算平台引入硬件安全芯片来保证计算机环境甚至网络环境的可信。这种可信是指环境中的软硬件实体按照预期的行为执行。国际上，安全芯片以满足 TCG (trusted computing group)规范的可信平台模块 TPM (trusted platform module)为主[3]，目前该规范已经升级到 TPM 2.0[8]。国内的安全芯片为可信密码模块 TCM (trusted cryptography module)[2]，其采用我国自主知识产权的密码算法 (如 SM2、SM3、SMS4 等)，并使用兼容我国公钥基础设施的双证书体系。

硬件安全芯片 TPM 和 TCM 拥有受保护的存储空间、隔离的执行环境和安全的密钥管理技术，能为计算平台提供认证、度量与报告和可信存储等安全功能。可信计算的安全特征已经被用于构建各种可信环境，提供基于硬件的安全防护，如可信 PC 平台[4]、可信云服务[9]和可信移动平台[10]等。特别是在云和移动环境中，安全芯片的存在形式出现了变化，如云环境主要以虚拟可信平台模块 vTPM 作为虚拟机的信任根，移动设备使用 ARM TrustZone 等安全扩展技术来提供隔离执行环境。TEEM[11]基于 ARM TrustZone 构建了一个面向用户的便携式可信模块，能为移动设备和传统 PC 等多种平台提供可信服务。MTA[12]使用移动设备的功能对可信计算的远程证明功能进行了改进。cTPM[13]对 TPM 2.0 的信任域进行了扩展，使用云技术增加了安全存储的空间，并能在用户多个设备之间进行安全共享。Chen[14]为 TPM 2.0 设计了灵活的签名原语，能实现各种签名机制，如传统的 Schnorr 签名、直接匿名签名和假名系统 U-Prove 等。

2.2 可信度量技术研究

终端平台的信任构建主要包括可信引导和完整性度量 2 个过程。可信引导[3,4]是指以 TPM/TCM 安全芯片为信任根，逐级度量系统启动过程中的硬件、BIOS、Bootloader 和操作系统内核；完整性度量是指操作系统内核启动后，对所有应用程序进行度量和保护，收集系统的各种配置信息。这 2 个过程构成了平台启动后生命周期的完整性信任链，能反映终端平台环境的信任状态，通过远程证明技术[4]，该信任状态可以传递给外部实体，从而建立网络环境中各个实体间的可信。

最早的完整性度量技术是 IBM 实现的 IMA 架构[5]，其使用 LSM hook 对 Linux 操作系统内核进行了扩展，使其能度量和记录所有可执行程序，并使用 TPM 对度量记录进行安全保护。IMA 是应用最广泛的完整性度量架构，目前已经被引用近 1 000 次，不过其在内核中没有实现对系统的管控，而且使用的技术也无法直接应用到 Windows 操作系统内核。后来 IBM 提出了 IMA 的改进版本 PRIMA[6]，只是通过将度量与信息流访问控制模型相结合来精简度量对象，提高系统效率。

M Nauman[15]等对 IMA 架构进行了移植和改进，使其能为 Android 智能手机平台提供度量服务和远程证明支持，并在 Linux 内核中模拟实现了一个 Mini TPM Emulator，提供基本的扩展和认证服务。Zhang[16]也提出并实现了一种简单高效的针对

移动平台的完整性保护解决方案，通过强制访问控制对可信与不可信区域信息流进行了严格的限制，可以防范大量恶意程序和攻击。CMU 学者对一些拥有独立处理器的外设进行了度量和证明机制的设计与实现，如对简单外设键盘的度量[17]和复杂外设网卡的度量[18]，由于外设的存储和计算能力受限，这些度量方法主要采用 HMAC 的方式对外设的所有内存和 I/O 进行散列计算，防止对外设的软件攻击行为。另外，类似的度量思路已经扩展到一些低端嵌入式设备，如 SMART 度量机制[19]主要采用 ROM 中固定的度量代码对需要验证的应用代码和数据进行 HMAC 计算。

3　安全 Windows 平台方案

本文专注于通过完整性度量技术设计安全终端，主要面向 Windows 平台，并采用自主 TCM 安全芯片提供硬件保护。本节首先给出一些特定 Windows 应用场景的安全需求，并制定安全目标；然后根据目标，对系统进行详细的设计。

3.1　需求与目标

微软已经宣布结束对 Windows XP 系统的支持，并不再提供任何技术帮助和安全更新服务[注1]。但是市场报告[注2]显示全球 37.2%的个人电脑仍在使用 XP 系统，特别是在一些国企、金融领域和政府等部门，XP 系统的应用比例高达 70%以上。如在自助服务终端、工业控制、电力控制等系统中，一些关键服务都运行在 Windows XP 系统上。在缺少微软的安全支持情况下，这些电脑将面临前所未有的安全威胁。以银行自动取款机 ATM (automatic teller machine)为例，目前很多都基于 Windows XP 系统，更新这些系统需要很大的成本，而且可能导致现有的软硬件不兼容，影响金融系统的可用性和稳定性；如果不更新系统，用户和企业将面临数据丢失、资金被盗等安全风险。

本文针对的应用场景如图 1 所示。体系中有多台终端，这些终端一般处于无人监控状态，并且运行一些安全系统，如 ATM 机；每个终端通过专用网络（如银行金融网络）与中心服务器（如银行服务器或者数据中心）相连。这些终端一般运行的软件服务比较固定，而且一般很长时间不会更新；终端操作的数据都会响应到中心服务器，这些数据（如银行卡密码等）的安全性必须得到严格的保证。例如通过篡改终端软件和各类普通的输入、输出硬件设备，恶意用户可能上传虚假的数据到中心服务器，也可能盗取用户的私密信息。

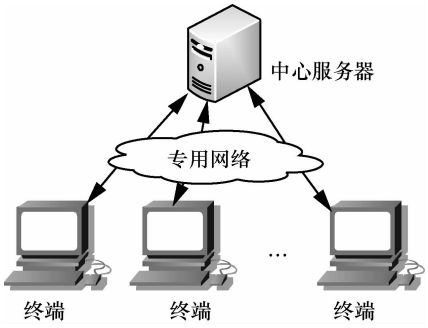


图 1　基本应用场景

基于这类应用场景的安全需求，本文抽象出了如下几个设计目标：

1) 保持终端设备的操作系统和用户体验不变；

2) 终端设备的身份能够得到验证，保证只有在合法终端上操作的用户数据才有效；

3) 终端设备的完整性能够得到验证，只有拥有预期配置的终端才能正常使用；

4) 终端设备运行的软件能够被管控，只有指定的软件能在终端上运行，其他软件都被禁止运行，能抵抗零日攻击等软件攻击行为。

3.2　系统设计

为了满足设计目标，本文以防篡改的 TCM 安全芯片为基础，通过扩展 Windows 内核模块，建立 Windows 终端平台的安全和可信。本文机制针对的应用场景主要是一些国有的关键业务和服务系统，因此采用自主的 TCM 安全芯片和国有密码算法。

图 2 给出了安全 Windows 平台的整体架构，主要由终端和服务器 2 部分构成，与图 1 的应用场景保持一致。终端实现了一个安全 Windows 平台，主要包含如下几个模块。

1) 可信密码模块 TCM：防篡改的平台信任根，主要提供安全存储（如平台配置寄存器 PCR）和安全密钥（如平台身份密钥 PIK 和平台加密密钥 PEK），能够引导平台可信。

2) Windows 内核层扩展：度量模块负责对平台加载的可执行程序、内核模块、动态链接库和其他配置文件等进行完整性度量，生成度量列表并扩展到 TCM 的 PCR 中安全存储；管控模块将度量模块

注1　http://windows.microsoft.com/zh-cn/windows/end- support-help

注2　http://arm.vrv.com.cn/index.php?m=apply&f=pbg

的完整性结果和白名单对比，决定是否允许软件运行；通信模块负责内核层扩展与上层安全服务的数据交互。

3) 安全服务：位于用户应用层，一方面与内核层扩展通信，负责获取度量列表中的平台完整性数据和进行白名单的设置；另一方面基于 TCM 与远程服务器进行平台身份和完整性认证。这 2 个通信过程都要在安全通道（如 SSL 中）完成。

服务器的认证服务负责验证各个终端平台的合法性，可信数据库为认证服务提供标准参考数据；管理员通过管理界面或者接口可以对可信数据库进行管理和维护。

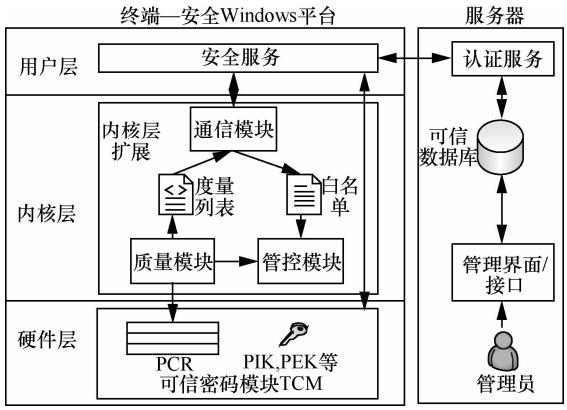


图 2 安全 Windows 平台整体架构

根据可信计算规范[2,3]，终端平台启动后通过 TCM 进行可信引导，当信任链传递到内核后，内核层的扩展模块开始接管系统，进行 Windows 平台的完整性度量。一个新的终端投入使用时，需要进行平台注册，确定其身份和预期运行的软件，并为之生成白名单。新终端注册并投入使用后，需要进行平台管控，只有白名单上的软件才能在该合法终端上运行。每次终端与服务器进行交易时，需要进行平台证明。当终端需要安装新的服务软件（较少发生）时，需要重新进行白名单的制定，完成平台更新。下面分别给出这几个阶段的详细设计。

3.2.1 平台完整性度量

可信计算以安全芯片为信任根，从 BIOS、Bootloader 到操作系统内核，一级度量一级，形成一条信任链，这就是可信引导过程，其研究已经很成熟。本文的完整性度量是操作系统内核启动后，通过扩展的 Windows 内核模块对加载到内存中的内核模块(.sys)、可执行程序(.exe,.com,.bat,.cmd)、动态链接库(.dll)和其他配置文件进行度量。

为了唯一标识一段特定的可执行代码，对其加载的内存镜像执行 SM3 密码杂凑算法。生成的 256 bit 杂凑值可以精确的标识该代码。不同的执行代码（如不同类型、不同版本、被篡改的代码等）都会拥有完全不同的杂凑值。扩展的内核度量模块会对 Windows 内核中所有加载的可执行代码进行 SM3 运算，并将这些杂凑值记录在内核的度量列表中。在平台注册和证明过程中，度量列表记录的杂凑值可以代表平台的完整性状态。如果度量列表被篡改，攻击者就可以隐藏自己的恶意代码进行攻击。为了防止这种恶意行为，使用 TCM 安全芯片的平台配置寄存器 PCR 来保证度量列表的完整性。

TCM[2]提供了 24 个平台配置寄存器，这些寄存器只能调用 2 个函数进行修改：1)每次系统重启时，对所有 PCR 进行重置清零；2)通过 TCM_PcrExtend 函数对指定的 PCR 进行扩展操作，即：新 PCR 值=SM3(原 PCR 值||SM3(可执行程序))。内核层度量模块对每个可执行程序进行 SM3 度量后，都会调用 TCM_PcrExtend 将杂凑值扩展到一个指定的 PCR 中进行保护。任何可执行程序在运行前，都会被度量并扩展到 PCR 中，恶意代码想要篡改 PCR 值，只能重启系统（重启后又会重新进入可信引导过程），而无法通过 TCM_PcrExtend 操作将 PCR 置为一个想要的结果（杂凑函数的不可逆性）。假设平台共加载了 n 段可执行代码 $p_1, p_2, \cdots, p_n$，这些代码的 SM3 杂凑值分别为 $m_1, m_2, \cdots, m_n$；那么，记录在度量列表 ML 中的每条记录为 $ML_i=<i, p_i, m_i>$，而扩展后的 PCR 值为 SM3(…SM3(SM3(0 || m_1) || m_2) … || m_n)。当认证服务接收到代表平台完整性状态的度量列表时，会以同样的方式（记为 Hash_Aggre (ML)）计算出杂凑聚集值 Ag

Ag=Hash_Aggre(ML)= SM3(…SM3(SM3 (0||m_1) ||m_2) …|| m_n)

ML= <ML_1, ML_2,…, ML_n>（一个长度为 n 的有序列表）

因此对度量列表的任何篡改（如改变加载的顺序或者任意一条记录的度量值）都会使杂凑聚集值与 PCR 值不一致，而无法通过服务器的认证。

3.2.2 平台注册

一台新的终端 (如新安装了一台 ATM 机)在投入使用前，必须得到中心服务器的认证。认证主要包含 2 个方面：对终端平台身份和完整性的认证。

平台完整性的认证可以使用以上完整性度量的结果，那么平台身份认证如何进行？另外，认证通过后，中心服务器会根据终端的完整性状态制定白名单，以便平台注册后终端能根据白名单进行自动管控。那么如何保证白名单的传输和存储安全？

本文使用 TCM 提供的安全密钥来解决上面的 2 个问题。TCM 安全芯片的唯一标识为密码模块密钥 EK，一般由厂商生成并固化在 TCM 内部，其私钥不会泄露。EK 可以用来建立平台所有者和平台身份。建立平台所有者时会创建存储主密钥 SMK，所有者和 SMK 的授权数据都通过 EK 公钥加密后植入 TCM 中，只有 TCM 内部能解密并存储这些授权数据。

在所有者授权下，TCM 可以采用 EK 建立平台身份密钥 PIK 和平台加密密钥 PEK。PIK 由 TCM 内部生成，并由一个可信方进行签署（即获得 PIK 证书），确保其可信性，可以用于对 TCM 内部信息进行数字签名，实现平台身份认证；PEK 由可信方（如密钥管理中心 KMC）生成，并通过数字信封的形式发送到 TCM。引入平台加密密钥的主要目的，是使 TCM 能够融入我国已有的公钥基础设施规定的签名和加密密钥双证书体系。基于 PEK 机制，用户可将自身的密钥托管于可信第三方，既便于丢失密钥之后的数据恢复，也利于国家对关键信息的管理控制。关于 PIK 和 PEK 的详细生成过程可以参考 TCM 规范[1]，本文不再累述。

本文使用 TCM 的 PIK 进行终端平台的身份认证，使用 PEK 来加密保护白名单。假设系统初始状态是完好的，即新安装的终端系统初始时可信；服务器从可信方获得了新终端平台的 PIK 和 PEK 公钥证书，即 CertPIK 和 CertPEK，PEK 已经发送给了新终端的 TCM。具体平台注册协议如表 1 所示(记新安装的终端为 Client，终端安全服务为 Client-S，服务器认证服务为 Server-V)。

表 1　　新 ATM 终端平台注册协议

终端	协议
1) Client	物理现场安装 ATM 软件和完整性度量软件，并重启运行 Client-S
2) Client-S→Server-V	注册请求 Req
3) Server-V→Client-S	随机挑战值 nonce
4) Client-S	a) 激活并加载 PIK 私钥 PriPIK 到 TCM； b) 载入 PEK 到 TCM，并与当前 PCR 值绑定； c) 生成 TCM 签名 Q = $\text{TCM\_Sign}_{\text{PriPIK}}${PCR, nonce}； d) 从内核读取度量列表 *ML*
5) Client-S→Server-V	*Q*, *ML*
6) Server-V	a) 验证 Client 平台 CertPIK 和 CertPEK； b) 计算聚集值 Ag= Hash_Aggre (*ML*)； c) 验证签名 $\text{Verify}_{\text{PubPIK}}(Q,Ag,\text{nonce})$； d) 验证 *ML* 中的每条完整性记录(与可信数据库对比)； e) 所有验证通过，保存 Client 终端信息到数据库，并基于 ML 生成白名单 *wl*； f) 为验证通过的 Client 终端生成服务密钥 *sk*； g) 使用 PEK 加密 *wl* 和 *sk*：ew= $\text{Encrypt}_{\text{PubPEK}}$(*wl*), esk = $\text{Encrypt}_{\text{PubPEK}}$(*sk*)
7) Server-V→Client-S	验证失败返回 Fail；或者验证成功返回加密分组(*ew*,*esk*)
8) Client-S	存储加密的白名单和服务密钥

终端平台安全服务接收到白名单密文 *ew* 后，只有在 TCM 内部使用 PEK 私钥才能解密，而且 PEK 在平台注册时与特定 PCR 值进行了绑定，如果平台配置发生变化（如执行了恶意软件），度量模块会扩展新的 PCR 值，导致 PEK 无法解密。这样可以防止平台注册后 TOCTOU 攻击[20]的发生。终端解密 *ew* 获得 *wl* (*wl*=$\text{Decrypt}_{\text{PriPEK}}$(*ew*))，通过通信模块写入内核的白名单 whitelist，并通知内核管控模块开启管控模式。在管控模式下，对于所有加载的程序代码，经过度量模块度量后，会将杂凑值交给管控模块处理，管控模块通过对比白名单，只允许杂凑值在白名单中的代码执行。这样任何不在白名单中的程序都会被阻止运行，可以抵抗零日攻击(zero-day attack)等安全威胁。

在管控模式开启之前，内核的白名单为空，因此只度量（重复的杂凑值不会被记录到度量列表中）不检验，称之为度量模式。对于一段特定的可执行代码 p，其在度量模式和管控模式下的处理流程分别如图 3(a)和图 3(b)所示(*ML* 为度量列表，whitelist 为白名单)。终端平台重启时，内存中的度量列表和白名单都会清空，在度量模式下启动到预期的 PCR 配置后，可以重新解密白名单并开启管控模式。

3.2.3　平台管控

新终端安装完成并且执行平台注册后，需要投入使用。在本文的场景中，除极少发生的特殊情况（如系统更新与升级），终端的软硬件配置不会改变。如果检测到软硬件的变动，很可能是攻击行为导致。在平台注册后，为了让终端平台能自主抵御一些攻击行为，特别是恶意软件的执行，本文设计了平台管控功能。

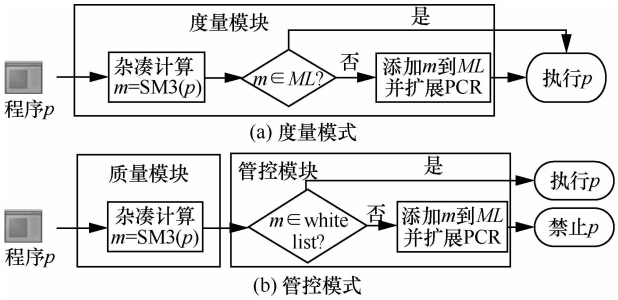


图 3　内核下的 2 种执行模式

3.2.4　平台证明与更新

从上可知，终端平台投入使用后都会在管控模式下运行，可以保持终端平台只运行预期的软件服务，防止未知或者恶意代码的执行。当终端与中心服务器进行数据交易时（如 ATM 交易时验证银行账户信息等），可以采用平台注册时的服务密钥来加密保护交易数据。同白名单一样，服务密钥由 PEK 加密保护，即 $esk = \mathrm{Encrypt}_{\mathrm{PubPEK}}(sk)$，只有当平台配置 PCR 与 PEK 绑定的状态一致时，TCM 芯片才能使用 PEK 解密获取 *sk*。服务密钥使用完后，安全服务会自动从内存中清空 *sk* 使用的痕迹，保证 *sk* 使用后不会泄露。

通过使用服务密钥，终端平台只需要注册一次，就可以进行多次交易和平台证明。不用每次交易时，都需要重新验证终端的身份和完整性状态。首先，能解密 *sk* 的 PEK 与 PIK 是成对出现在同一个 TCM 芯片的，可以确定终端的合法身份；其次，能解密 *sk* 需要终端当前配置与预期一致，可以确定终端的完整性状态；最后，*sk* 加密数据可以保证交易的机密性。

当终端需要进行软件更新时（很少发生），需要管理员将终端的监控模式切换到度量模式，并安装新软件后重新进行平台注册过程。重新注册后，原先的白名单和服务密钥都失效，终端会根据新的白名单进行管控。注意，在管控模式下，新的服务软件由于不在原先白名单下，故无法进行安装和运行，因此需要先切换到度量模式。该切换过程只能由指定管理员进行，否则如果允许随意切换模式，终端就会失去自主管控的功能。管控实际上也是一种基于白名单的强制访问控制方法。

4　安全 Windows 平台实现

本节介绍在 Windows XP 系统上对以上安全终端平台方案的原型实现。实现的难点主要在对 Windows 操作系统内核的扩展，安全技术主要体现在对 TCM 安全芯片的使用；另外，对终端的安全服务以及服务端的认证服务和管理界面也进行了完整的原型实现。表 2 给出了相关的编程语言及代码量评估。

表 2　安全 Windows 平台实现的代码量

系统组件	模块功能	代码量/行	总计
终端—内核扩展	Hook 功能	38(asm)+169(C)	38 行汇编和 2 202 行 C 代码
	SM3 杂凑	293 (C)	
	散列表	155 (C)	
	度量模块	232 (C)	
	管控模块	167 (C)	
	通信模块	204 (C)	
	其他	982 (C)	
终端—安全服务	TCM 交互模块	1 012 (C)	1 791 行 C 代码和 5 946 行 Java 代码
	内核交互模块	779 (C)	
	通信与管理界面	5 946 (Java)	
服务端	认证服务	1 404 (C)	1 404 行 C 代码，8 372 行 Java 代码和 7 200 行 jsp 代码
	管理界面	8 372 (Java) + 7 200 (jsp)	

4.1　系统配置与安全芯片

目前支持 TCM 的安全 PC 主要有联想的 ThinkCentre M8000t、M4300t 系列[注1]和同方的超翔 Z 系列[注2]台式机。这些安全终端的 TCM 一般通过 LPC (low pin count)接口集成在主板上。为了支持更多平台，采用厂商提供的 PCI (peripheral component interconnect)接口的 TCM 芯片[注3]。目前终端平台的主板上都带有 PCI 插槽，将 TCM 芯片直接连接 PCI 插槽，安装相应的驱动后可以使用其安全功能。目前访问 TCM，主要通过可信软件库 TSM[2]提供的接口进行，厂商并没有公开驱动层调用的接口；因此，本文实现与 TCM 交互的部分时，都由终端用户层的安全服务完成。不过，文献[11]中基于 ARM 开发板对 TCM 功能进行了模拟实现，并用 USB 接口连接 ARM 开发板和主机，使其可以代替 TCM 安全芯片使用。该 TCM 模拟开发板可以提供驱动层调用的接口，这种方法主要供研究使用。本文实现的安全 Windows 系统在这些终端机器（LPC 的 TCM、PCI 的 TCM 或者 ARM 模拟的 TCM）上都

注 1　http://appserver.lenovo.com.cn/Lenovo_Series_List.aspx? Category-Code=A02B30C01

注2　http://www.tongfangpc.com/products/chaoxiangz/index. html

注3　http://www.tsinghuaic.com/pdf/同方微电子可信计算解决方案.pdf

能正常运行，可以满足各种使用场景。

4.2 Windows 内核扩展

对 Windows 内核的扩展主要通过 NT 式驱动程序[21]实现。Windows 引导过程大致可以分为 3 步[22]：加载内核模块，内核的初始化，会话创建和用户登录过程。内核模块加载时的度量主要由可信引导过程进行，而驱动程序主要在内核初始化时进行加载，加载后会对所有可执行程序进行度量。那么，如何在内核中插入度量点以及实现管控功能？

度量应该在程序执行之前进行。Windows 加载所有的可执行模块（包括 exe、dll 和 sys 文件等）时都要通过内核函数 NtCreateSection 生成一个内存区对象（section object）映射进内存[23]。通过挂钩(hook)该函数，可以截获所有将要加载的可执行模块，截获后可以通过 SM3 对内存区对象进行杂凑计算并扩展到 TCM。Hook 技术有很多种，如 SSDT Hook、Inline Hook、IDT Hook 等[23]。采用 SSDT Hook 技术对 NtCreateSection 函数进行了挂钩，并以此作为度量点(38 行汇编和 169 行 C 代码)。

为了实现度量功能，在内核层实现了 SM3 算法(293 行 C 代码)，度量模块使用该算法进行杂凑计算，管控模块将其输出结果与白名单对比。在内核中，需要反复使用查询（或者检索）操作(见图 3)，如：一个可执行程序可能被加载多次，对于没有被篡改的重复程序，不会重复度量，需要查找度量列表来检查重复性；另外在管控模式下，每次度量都要检索白名单，决定是否允许执行。这些操作频繁发生，如果采用线性查找，那么会影响系统性能。本文采用散列表的方法提高查找速度，通过关键值进行索引，可以以接近 $O(1)$的时间复杂度完成查找操作。

Windows 内核处于多线程模式，记录的度量值可能与捕获的可执行程序不匹配。为了实现度量记录的完整性，采用自旋锁对度量列表（实现为内核的一个链表结构）进行了保护。为了在度量模式和管控模式之间进行切换，内核还维护了一个比特位进行标记。为了与上层安全服务进行数据交互，内核定义并实现了多种 DeviceIoControl 通信方式，使

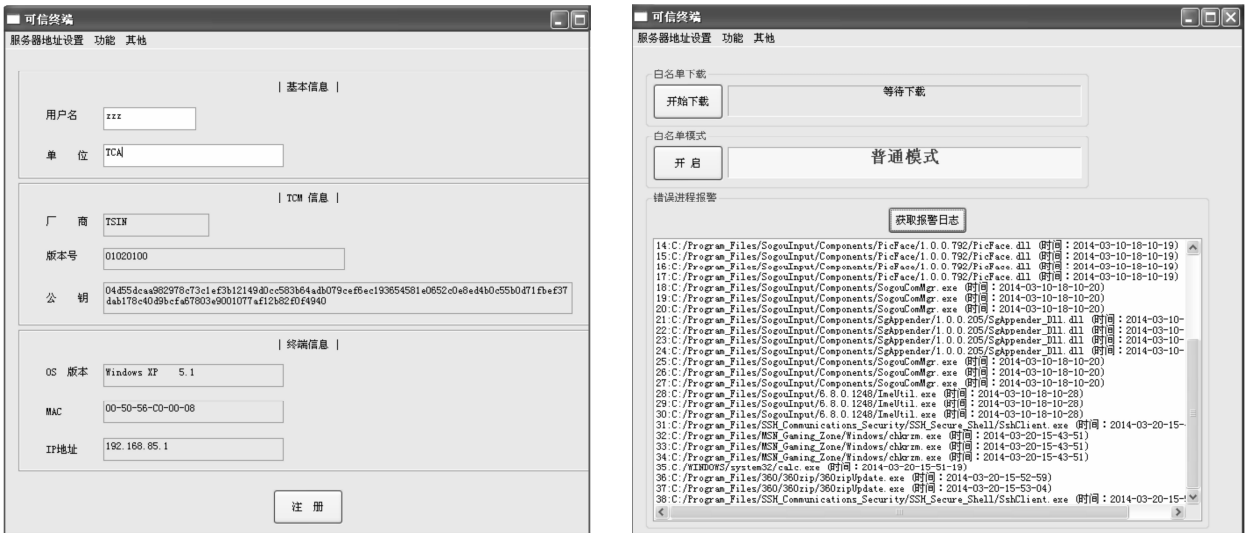
```
…
8340f78bad23830f986bd6f09c3acd8879294d9e8723525320e02ac08562ccf8 netapi32.dll
c2ad5009b0b7d99df78592146afa4ebd8ce4072a829e64bf3cfdd052b6ceb7e2 wintrust.dll
6710f42571780850b32228cd457ac5ccf978a7db4a6f79a5b5bfb348b465245a win32k.sys
1d973ba81f1983b71ddefd54f3c8953fec72f1754803fc6cc2980fa12922d2ea authz.dll
f3bd9a21b683dda082adee9bd23afa7687d15023f349b860455b8632dafba306 nddeapi.dll
52584c824a92f342e0b800dc7b8eb893ee6081b47e9e22288d03b727903107a7 profmap.dll
705b6e45db436a3fc6de325cf8835e4dee2e4bb5a1a8a359ca1013e8cfd16a5c psapi.dll
b1460f8aca8087a5ec065d9c43088b311af6leld89df6953d787ad59762f9f50 regapi.dll
8751a31d6de2d44eb21b6fcd5b9c8794eede7e5e74e09739c5761a0ab3c6533f setupapi.dll
48ec9512c28e910fd2bf0daed7fd9e82dcc238a98f9ee0773072be45e1ba616e winsta.dll
ee65c46a1df94cc45e3c34f62f375eb8ca3ccbb7d9265dbe08cd74a8afc 14237 ws2_32.dll
…
036ab6e8948dfcdae68c2b77ee2fc67fc876a3773441730cd2653379958e 6ee3 odbcint.dll
88b93caf0cf1686268691fe44ad3a31a9a0d5b296af1a9d9flb349057c61bfa4 shsvcs.dll
864e208fc49ab15511db412b37cfd12a3df067bc54c1cf77dab73bdaf2d65c8d sfc.dll
1bc9550069666414b560dc850ba4c1c4c05cfa2b52593b744b3c6e04d3ae8ece sfc_os.dll
2deabb8ad0b019976d7d9c3al f2c7793935afa7c2728adcdf803ba8966269adb services.exe
1da5f7d85e6d544f75e64b1d3e4889f2c94260a8a1ffc4fcc4f5b9d78ccbe290 lsass.exe
…
```

(a) Windows XP系统启动过程

```
…
b283c0f91ef5762e1fc99696d669c9b6302bc046d431aca4acd7176912e5f21e myeclipse.exe
37cbaa2d9cd4df85d1a554a8432e52d594f644de9a3455373e673d29313bce76 eclipse_1206.dll
4c857e6a1d2730040fa4fecf5580ccc2bcd8b32433aefebc657c24df80a2b721 jvm.dll
7368598b2dfc8cb07b382a1e51226b12261658346c2e5271c6a1198fbd5b78cc hpi.dll
353a16c6a90f2673304606f1c378585bedab70969d5f5104ellafaef507635fe verify.dll
ed5elcfad0709cf9a74ca785ac1b176f9e583f3d965f5bb20b7029e201082164 java.dll
f080b53763bb8819f6dec2caacb26bead8884d1ffd95307f11f894484bd2b3f9 zip.dll
48c2adf6a592c192d573c097baeb8b36928dc068565f286e543a6b011c0cddc9 net.dll
66b57257336531e6f8ef79b138619812518b0cd41af1bfd255dd1c33b86dfl cf nio.dll
99bb102867bfd805df943db05b6b93fd5f838c8e854e33f3c15a0ac37bc4b5fa swt-win32-3557.dll
fe10c80e831d455c32204dd7c9c78005502e2904f25d1596c891dcfaf58731e2 jwinHttp-1.0.0.dll
8f0b1bb7b13b02966ec36a2d8f384e20832e896004c1def6ad5562aac806beea localfile_1_0_0.dll
…
```

(b) 系统启动tomcat服务器以及myeclipse软件后

图 4　度量日志



（a）平台注册 （b）平台管控

图 5 终端安全服务界面

上层安全服务能获取度量日志、设置白名单和切换内核模式等操作。

图 4(a)给出了 Windows XP 系统启动过程中捕获的部分度量日志，图 4(b)给出了系统启动 tomcat 服务器以及 myeclipse 软件后，内核中度量模块捕获的度量日志。度量日志记录主要包括文件名和 SM3 杂凑值。

4.3 服务与管理功能

服务功能包含 2 大块：终端的安全服务和服务器的认证服务。终端安全服务功能包含 3 个方面：管理和使用 TCM 的功能(1 012 行 C 代码)；与内核交互(779 行 C 代码)；与服务器认证服务交互以及为终端管理员提供操作的 GUI(5 946 行 Java 代码，GUI 通过 swt 实现)。TCM 相关的功能通过调用 TSM 软件栈[2]实现，主要包括：创建所有者(Tspi_TCM_TakeOwnerShip)、生成平台密钥(Tspi_Key_ CreateKey)、密钥加载(Tspi_Key_ LoadKey)、签名操作(Tspi_Hash_Sign)、解密操作(Tspi_Data_Decrypt)和 PCR 相关操作(Tspi_TCM_PcrExtend 和 Tspi_TCM_PcrRead)等。TCM 以及内核相关的操作都实现为 dll 库，GUI 通过 Java swt 框架实现，通过 jna 技术与 dll 连接，将管理功能和完整性数据显示在界面上，如图 5 所示。

服务器的认证服务主要是验证终端的身份和完整性数据，并对比数据库后返回加密的白名单和服务密钥，采用 C 实现，共 1 404 行代码。服务器的管理界面供管理员操作可信数据库，如参与白名单的审核与制定、查看已经注册有效的终端平台等。数据库采用 mysql，与数据库交互采用 Java，Web 页面显示采用 jsp 实现，如图 6 所示。

图 6 完整性管理界面

5 系统评估

与存在的纯软件安全工具相比，本文方案存在如下几个特点：1) 使用 TCM 的平台配置寄存器保证度量日志的完整性；2) 使用 TCM 的平台身份密钥证明终端平台身份的合法性；3) 使用 TCM 的平台加密密钥保护白名单不被篡改；4) 支持度量（被动监控）和管控（主动防御）2 种内核扩展模式。这些特征都基于 TCM 的硬件保护能力，能抵抗各种软件攻击。

5.1 安全性分析

白名单攻击：很多防病毒软件(如 Bouncer 和

Anti-Executable [注1])也采用白名单机制，但是白名单本身的安全性却很少被考虑，如果白名单被篡改，防御机制也就形同虚设。假设一台新安装的终端初始状态可信，并依此制定了白名单，并且制定过程由可信的服务器管理员进行，保证了白名单的初始可信性。白名单通过 PEK 加密后进行传输和存储，基于可信计算 PEK 的特性，只有特定的终端 PCR 配置和 TCM 安全芯片能解密白名单，可以保证注入到内核的白名单与初始可信状态一致，白名单注入内核后会立即自动开启管控模式，进入主动防御阶段。如果终端安全服务被攻破，其可能篡改内核的白名单；但是根据度量机制，非法的安全服务在执行之前，先被度量并匹配白名单，在管控模式下，该非法的安全服务无法执行。

TOCTOU(time of check to time of use)[20]攻击：可信计算的一个难题是 TOCTOU 攻击，即平台在证明时刻(time of check)完整性状态良好，但是证明过后投入使用时(time of use)完整性状态被攻击者破坏。本文在平台注册后，采用白名单进行了主动防御，使平台在使用过程中维持证明时刻的可信状态，可以有效防止 TOCTOU 攻击。

零日攻击：很多攻击者使用零日攻击漏洞在计算机系统中安装恶意软件或者后门，一般杀毒软件无法识别。而基于白名单机制，任何不在白名单中的程序加载到内存前都会被度量，并被阻止执行。即使系统存在一些未知的零日漏洞，恶意软件无法篡改白名单，也无法修改内核模式下的管控模式，因此也就无法对系统造成破坏。

重放攻击或者假冒攻击：恶意终端平台可能截获合法平台的注册信息，并简单地转发进行攻击，通过在平台注册信息中使用新鲜生成的 nonce 值，可以防止这种重放攻击。恶意攻击者可能假冒合法终端平台对认证服务器进行欺骗攻击，首先需要获取一个合法的身份（即 TCM 密钥），其次需要一个合法的完整性状态（PCR 值），而根据可信计算规范[2,3]，TCM 芯片私钥不会泄露，TCM 的 PCR 寄存器只能通过 reset 或者 extend 操作进行更改，因此，攻击者无法完成假冒攻击。即便一台拥有 TCM 的恶意终端，其 PIK 证书和 PEK 证书也无法通过认证服务器的检查；如果其使用其他 TCM 平台的 PIK 证书和 PEK 证书，其本身的 TCM 芯片无法解密白名单和服务密钥。

5.2　性能分析

本文设计的安全 Windows 平台核心功能都在终端实现，本节将从如下 3 个方面来评估系统性能：度量功能消耗的时间、TCM 命令消耗时间以及白名单查询消耗的时间。

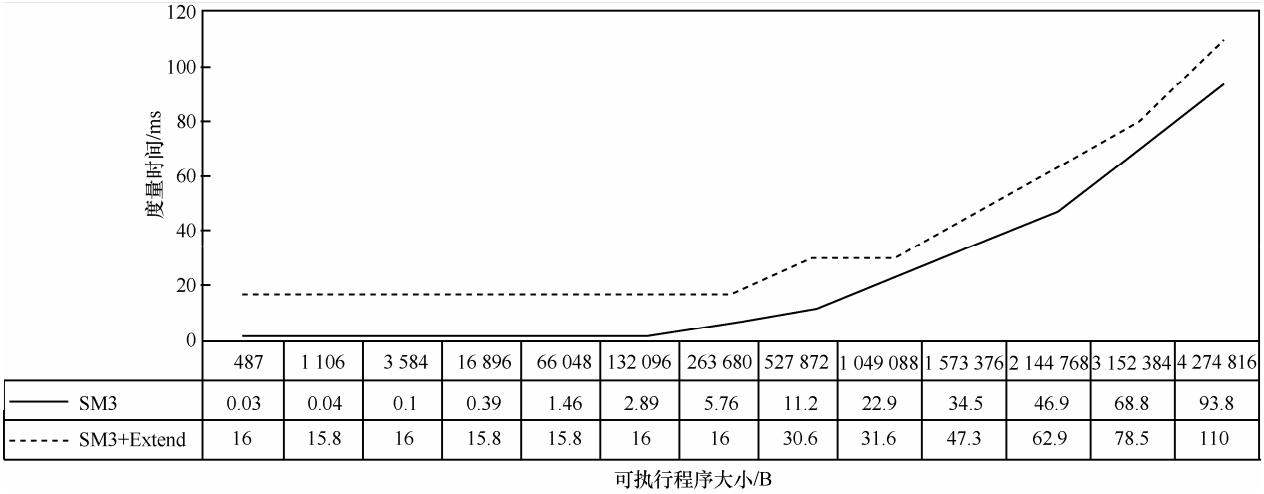
实验终端主机平台为联想 ThinkCentre M8300t，其配置如下：操作系统为 Windows XP SP3；CPU 为 i7-2600 @3.4 GHz；内存为 3.39 GHz 3 GB；安全硬件为同方微电子 PCI 接口的 TCM 芯片。经过评估，开机时度量耗时不超过 20 s，度量一个可执行程序平均时间 30 ms，主要 TCM 命令都在 100 ms 之内，白名单查询时间可以忽略不计（只需要几 μs），因此，本文设计的系统不会对 Windows 终端平台的性能造成太大负担，不影响正常使用。下面进行详细评估和分析。

1) 度量性能

主要评估了度量时间与可执行程序大小的关系（如图 7 所示）以及度量对 Windows 系统开机时间的影响（如图 8 所示）。在评估度量时间与可执行程序大小的关系时，选择了 13 个不同大小的可执行程序，最小的 0.5 KB，最大的 4 MB，对每个文件度量了 50 次（度量分为只进行 SM3 杂凑算法和同时进行 SM3 杂凑算法与 TCM 扩展操作），并选取其平均值，注意图中横坐标并不是线性增长的。在评估度量机制对开机的影响时，对 Windows XP 系统开机的可执行程序进行了收集，包括 Windows 系统程序 600 个和其他应用程序 400 个，以此为样本，使用度量算法（SM3+TCM 扩展）对开机时间进行了评估。对于每次选取的开机样本个数（从 200 到 1 000 不等），都进行了 10 次度量，并记录其平均值。

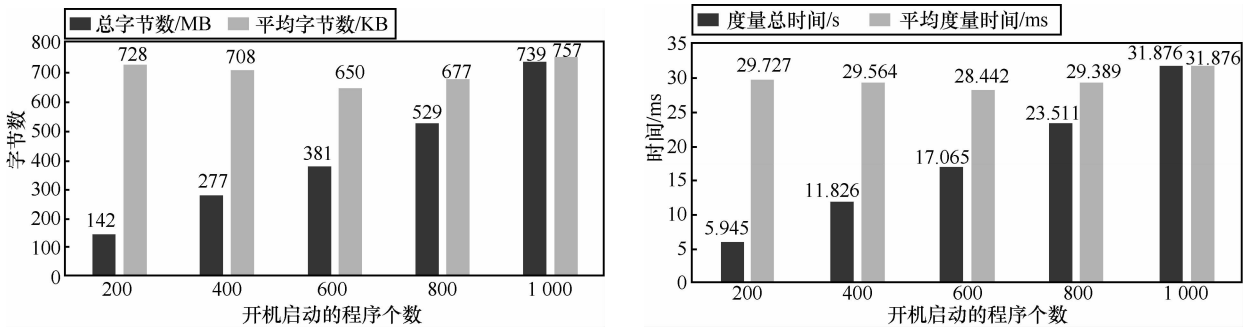
分析：从图 7 可知，只计算 SM3 时，度量时间与可执行程序大小几乎成线性关系增长；而计算 SM3 后并进行 TCM Extend 操作时，当可执行程序大小在 256 KB 之下时，由于 SM3 杂凑计算时间与 Extend 操作时间相比几乎可以忽略，因此主要时间为 Extend 命令的执行时间，在 15 ms 和 16 ms 之间；而当文件大小大于 512 KB 时，总体时间基本是 SM3 计算时间加上 Extend 时间，注意 Extend 时间

注1　http://security.zdnet.com.cn/security_zone/2010/ 0830/1866752.shtml



	487	1 106	3 584	16 896	66 048	132 096	263 680	527 872	1 049 088	1 573 376	2 144 768	3 152 384	4 274 816
SM3	0.03	0.04	0.1	0.39	1.46	2.89	5.76	11.2	22.9	34.5	46.9	68.8	93.8
SM3+Extend	16	15.8	16	15.8	15.8	16	16	30.6	31.6	47.3	62.9	78.5	110

图 7　度量时间与可执行程序大小的关系



(a) 开机度量规模　　(b) 开机度量时间

图 8　开机度量规模与时间评估

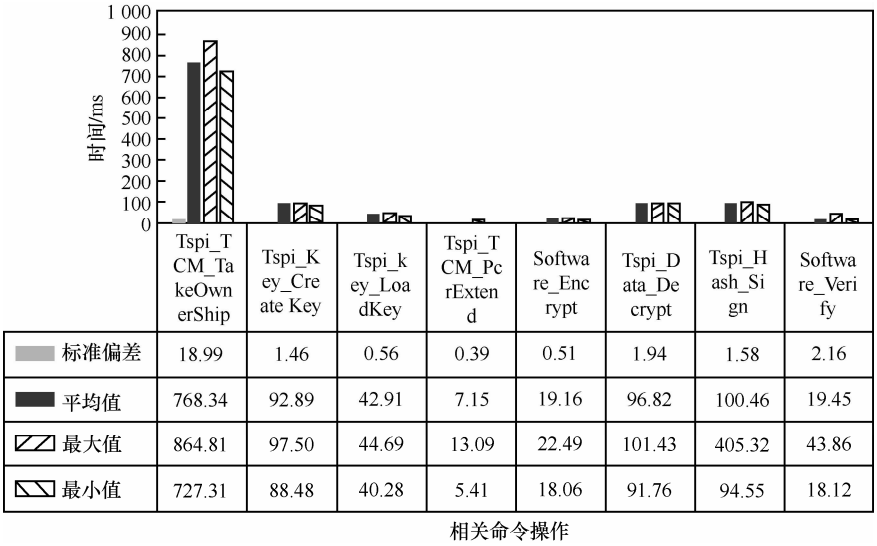
基本是固定的。大部分可执行程序大小在 1 MB 以下，故其 SM3 杂凑时间几乎可以忽略不计，主要由 Extend 的时间决定。

通过统计，大部分 Windows XP 开机后加载的可执行程序在 400~600 的范围内，图 8(b)给出了 200 到 1 000 个可执行程序度量需要的时间，度量时间包括 SM3 杂凑计算和 Extend 扩展的时间；图 8(a)给出了相应的度量规模（即度量的字节数）。前 600 个程序是收集的开机自动启动的可执行程序，大部分是 Windows XP 系统自带程序，可见平均长度和度量时间随着程序个数增加而有减少的趋势；后面增加 400 个程序是加载的其他应用程序，平均长度和执行时间比系统程序稍有增长。总体而言，从图 8 中数据可以看到，平均程序长度大概为 700 KB，平均度量时间大概为 30 ms，正常开机度量机制增加的时间消耗在 20 s 之内。

2) TCM 命令性能

主要评估了本文方案需要用到的相关 TCM 命令的时间，其中加密和验证操作是纯软件实现，不需要 TCM 参与，由服务器执行。实验评估时加密和验证也在终端上执行，具体相关命令的执行时间如图 9 所示。对于每个 TCM 命令，测试了 1 000 次，然后记录时间的最大值、最小值、平均值和标准偏差。加密和签名都采用 256 bit 的 SM2 算法。

分析：TakeOwnerShip 操作需要的时间（768 ms）比较长，不过其只需要执行一次即可，一个平台只能生成一个 Owner。平台身份密钥 PIK 主要由 CreateKey 创建，对于本文的系统，只需要一个 PIK 即可，故 CreateKey 命令也只需要执行一次。加载密钥 LoadKey、签名 Sign、解密 Decrypt 在每次平台注册时，都由 TCM 执行，大约需要 250 ms；验证签名 Verify 和加密 Encrypt 都由服务器纯软件执行，分别只需要 20 ms 左右时间。Extend 操作只需要（7.15±0.39）ms 的时间，而在度量性能评估时，其最大可能达到 20 多 ms，这主要由从 SM3 计算到 Extend 操作的一些中间计算操作影响。



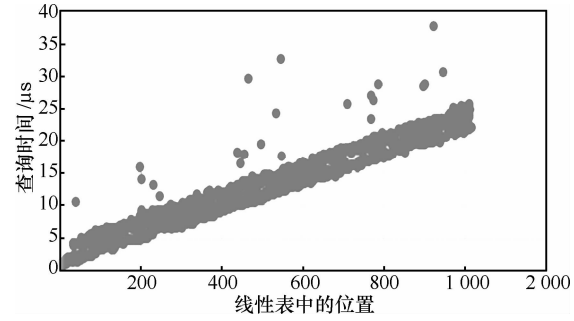
	Tspi_TCM_TakeOwnerShip	Tspi_Key_Create Key	Tspi_key_LoadKey	Tspi_TCM_PcrExtend	Software_Encrypt	Tspi_Data_Decrypt	Tspi_Hash_Sign	Software_Verify
标准偏差	18.99	1.46	0.56	0.39	0.51	1.94	1.58	2.16
平均值	768.34	92.89	42.91	7.15	19.16	96.82	100.46	19.45
最大值	864.81	97.50	44.69	13.09	22.49	101.43	405.32	43.86
最小值	727.31	88.48	40.28	5.41	18.06	91.76	94.55	18.12

图 9　TCM 命令执行时间

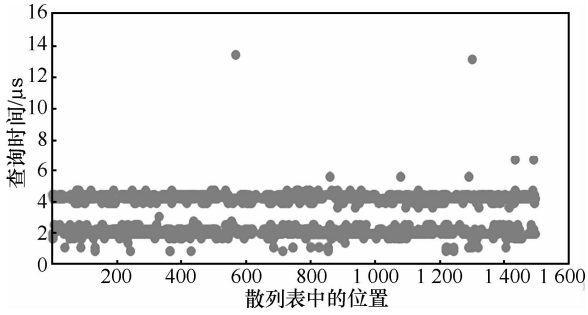
3) 白名单查询性能

白名单查询在内核管控模式下需要频繁使用，度量日志的查重操作也类似。这里主要比较基于线性表和散列表 2 种方式查询白名单的时间。实验中，收集了 1 020 条度量值，并将这些数据设置为白名单。线性表通过数组存储，数组空间为白名单规模，即 1 020；散列表预先设置了 1 500 个散列值的空间（散列表的空间应该大于等于白名单的规模，否则需要重新设置散列表）。对于每种方式，执行 1 000 次查询，每次随机选择白名单中的度量值作为查询的关键字，返回查询的时间以及关键字在表中的位置。图 10 中分别记录了白名单查询（1 000 次）在线性表和散列表下的散点。

分析：线性表的查询时间（如图 10(a)所示）与关键字在表中的位置有关，基本是线性关系，即复杂度为 $O(n)$。而散列表的查询时间（如图 10(b)所示）与关键字在表中的位置无关，图中时间主要集中在 2 μs 和 4 μs 这 2 根线上，由于散列算法中使用线性探测再散列的方式处理冲突，即当 i 位置已经填有记录时，则探测 i+1 的位置；2 根比较稳定的线，说明大部分查找只需探测 1 次或者 2 次可找到关键字，即复杂度为常量 $O(1)$。对于不在表中的关键字，散列表的查询时间是最小值（即探测 1 次即可），而线性表的查询时间是最大值（需要查询整个线性表）。显然对于数据量大且需要反复进行查询时，散列表方法更加优越。因此，本文系统采用散列表方法。另外，通过纵向比较，白名单查询时间（μs 级）与度量时间（ms 级）相比基本可以忽略，不会影响系统整体性能。



(a) 基于线性表



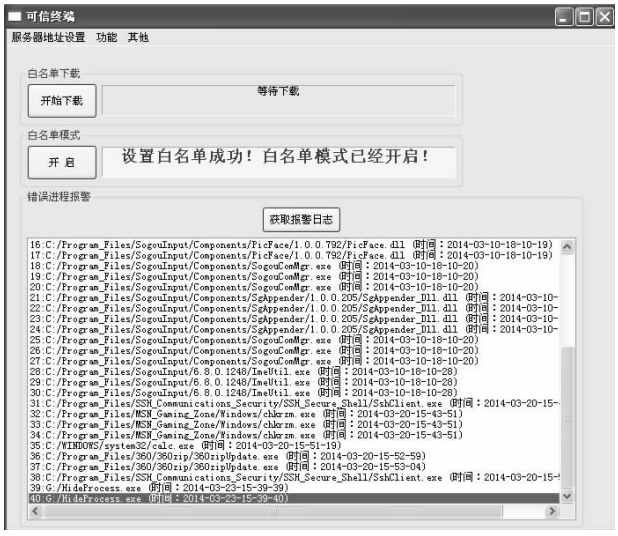
(b) 基于散列表

图 10　查询白名单的时间

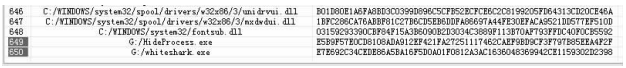
5.3　攻击实验

为了测试本系统的防攻击能力，测试了一个 U 盘自启动的病毒，双击 U 盘后会自动启动一个 HideProcess.exe（Rootkit）的进程，该进程会释放一个 whiteshark.exe 的病毒，并通过 hook 系统函数

NtQuerySystemInformation 对病毒进程进行隐藏。在管控模式下，HideProcess.exe 会被阻止运行，也就无法释放 whiteshark.exe，如图 11(a)所示；当关闭管控模式，在度量模式下 HideProcess.exe 和 whiteshark.exe 都会正常运行，但会被度量，如图 11(b)所示。



(a) 白名单模式（主动防御）



(b) 度量或者普通模式（被动监控）

图 11 Rootkit 检测与防御

6 结束语

本文设计并实现了 Windows 系统的可信度量技术，通过扩展 Windows 内核对系统关键 API 进行 hook，能度量系统加载的所有可执行程序；并基于 TCM 芯片证明终端平台的身份合法性以及完整性状态的可信性。本文的方案适合一些依赖 Windows 操作系统且环境比较固定的应用场景，如 ATM 机、自动售票系统等。

参考文献：

[1] Available online[EB/OL]. http://web.vrv.com.cn/news_detail/newsId=e7f4db6b-7321-4fc3-b475-86eed3e83377.html.

[2] 国家密码管理局. 可信计算密码支撑平台功能与接口规范[S]. 2007.
Chinese Commercial Cryptography Administration Office. Functionality and Interface Specification of Cryptographic Support Platform for Trusted Computing[S]. 2007.

[3] Trusted Computing Group. Trusted Platform Module Main Specification [S]. Version 1.2, Revision 103 2007.

[4] BRYAN P, JONATHAN M M, ADRIAN P. Bootstrapping trust in commodity computers [A]. Proceedings of the IEEE Symposium on Security and Privacy[C]. 2010.414-429.

[5] SAILER R, ZHANG X L, JAEGER T, *et al*. Design and implementation of a TCG-based integrity measurement architecture [A]. Proceedings of USENIX Security '04[C]. Berkeley: USENIX Association, 2004. 223-238.

[6] JAEGER T, SAILER R, SHANKAR U. PRIMA: policy-reduced integrity measurement architecture[A]. In Proceedings of the 11th ACM Symposium on Access Control Models and Technologies[C]. New York: ACM Press, 2006. 19-28.

[7] 冯登国, 秦宇等. 可信计算技术研究 [J]. 计算机研究与发展, 2011, 48(8): 1332-1349.
FENG D G, QIN Y, *et al*. Research on trusted computing technology[J]. Journal of Computer Research and Development, 2011, 48(8): 1332-1349.

[8] Trusted Computing Group. Trusted Platform Module Library: Part 1-Part 4 [S]. Family 2.0, Level 00 Revision 00.96, 2013.

[9] NUNO S, RODRIGO R, KRISHNA P. G, STEFAN S. Policy-sealed data: a new abstraction for building trusted cloud services [A]. Proceedings of the 21st USENIX Security Symposium[C]. Bellevue, WA, 2012.10.

[10] KURT D, JOHANNES W. Implementation aspects of mobile and embedded trusted computing [A]. Proceedings of the 2nd International Conference on Trusted Computing[C]. 2009.29-44.

[11] FENG W, FENG D G, WEI G, *et al*. TEEM: a user-oriented trusted mobile device for multi-platform security applications[A]. Trust and Trustworthy Computing[C]. 2013.133-141.

[12] FENG W, QIN Y, FENG D G, *et al*. Mobile trusted agent (MTA): build user-based trust for general-purpose computer platform[A]. Proceedings of Network and System Security[C]. Springer Berlin Heidelberg, 2013.307-320.

[13] CHEN C, HIMANSHU R, STEFAN S, ALEC W. cTPM: a cloud TPM for cross-device trusted applications[A]. Proceedings of 11th USENIX Symposium on Networked Systems Design and Implementation[C]. DEATTLE, WA, 2014.187-201.

[14] CHEN L Q, LI J T. Flexible and scalable digital signatures in TPM 2.0 [A]. Proceedings of ACM SIGSAC Conference on Computer and Communications Security[C].New York, NY, USA, 2013. 37-48.

[15] NAUMAN M, KHAN S, ZHANG X, SEIFERT J P. Beyond kernel-level integrity measurement: enabling remote attestation for the Android platform[A]. Trust and Trustworthy Computing[C]. 2010.1-15.

[16] ZHANG X W, JEAN-PIERRE S, ONUR A. Design and implementation of efficient integrity protection for open mobile platforms [J]. IEEE Transactions on Mobile Compuring, 2014, 13(1):188-201.

[17] LI Y L, JONATHAN M. M, ADRIAN P. SBAP: software-based attestation for peripherals [A]. Proceedings of the 3rd International Con-

ference on Trust and Trustworthy Computing[C]. 2010.

[18] LI Y L, JONATHAN M M, ADRIAN P. VIPER: verifying the integrity of PERipherals' firmware [A]. Proceedings of the 18th ACM Conference on Computer and Communications Security[C]. 2011.3-16.

[19] KARIM E D, AURÉLIEN F, DANIELE P, GENE T. SMART: secure and minimal architecture for (establishing a dynamic) root of trust[A]. Network and Distributed System Security Symposium (NDSS)[C]. 2012.

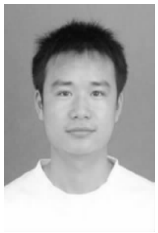
[20] SPARKS E R. A security assessment of trusted platform modules [R]. Technical Report TR2007-597, Dartmouth College, 2007.

[21] 张帆等. Windows 驱动开发技术详解[M]. 北京：电子工业出版社, 2008.
ZHANG F, *et al*. Windows Driver Development Internals[M]. Beijing: Publishing House of Electronics Industry of China, 2008.

[22] 潘爱民. Windows 内核原理与实现[M]. 北京：电子工业出版社, 2010.
PAN A M. Windows Kernel Principle and Realization [M]. Beijing: Publishing House of Electronics Industry of China, 2010.

[23] 谭文，邵坚磊. 天书夜读-从汇编语言到 Windows 内核编程 [M]. 北京：电子工业出版社, 2008.
TAN W, SHAO J L. Reading Sanscrit at Midnight – From Assembly Language to Windows Kernel programming [M]. Beijing: Publishing House of Electronics Industry of China, 2008.

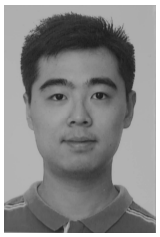
作者简介：



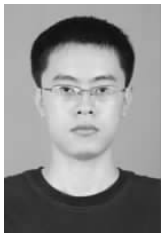
冯伟（1986-），男，湖北荆州人，中国科学院软件研究所博士生，主要研究方向为可信计算、网络与系统安全。

秦宇（1979-），男，重庆人，中国科学院软件研究所高级工程师，主要研究方向为可信计算、网络与系统安全。

冯登国（1965-），男，陕西靖边人，中国科学院软件研究所研究员，主要研究方向为可信计算与信息保障、网络与信息安全。



杨波（1988-），男，河南沁阳人，中国科学院软件研究所博士生，主要研究方向为可信计算、移动平台匿名系统。



张英骏（1990-），男，河北秦皇岛人，中国科学院软件研究所硕士生，主要研究方向为可信计算与信息保障。