doi:10.3969/j.issn.1000-436x.2014.z2.011

跨平台的可信执行环境模块方案研究

张倩颖，赵世军，冯伟，秦宇，冯登国
(中国科学院 软件研究所，北京 100190)

摘 要：针对现有 TPM、MTM 等可信计算模块不能跨平台使用，未考虑算法、协议、功能更新等问题，提出一种基于硬件的可信执行环境模块（TEEM, trusted execution environment module）架构，该架构利用 ARM TrustZone 技术构建一个运行在硬件安全隔离环境中的可信计算模块。该模块能够为多种平台提供可信计算功能，具备较强的移动性和便携性，并且允许用户根据需要灵活地配置、升级模块的功能和算法。设计并实现了基于 TEEM 架构的原型系统，原型系统的安全性分析和性能测试结果表明，TEEM 能够为用户提供一个安全、稳定、高效的可信执行环境。

关键词：可信执行环境；可信计算；ARM TrustZone；可信平台模块；移动可信模块
中图分类号：TP309 **文献标识码**：A **文章编号**：1000-436X(2014)Z2-0072-14

Research of a trusted execution environment module for multiple platforms

ZHANG Qian-ying, ZHAO Shi-jun, FENG Wei, QIN Yu, FENG Deng-guo
(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: The current TPM, MTM and other trusted computing modules don't take into account the variety of platforms and the update of the inside algorithms, protocols and functions. A hardware trusted execution environment module (TEEM) architecture, which uses ARM TrustZone technology to build a trusted computing module running in a secure isolated environment is designed. Proposed module not only supports variety of platforms, but also has strong mobility and portability. Moreover, it allows configuring and updating functions and algorithms of the module flexibly. A prototype system is implemented and its performance is tested. By analyzing the security of the system and the measurement results, it is shown that TEEM provides users with a safe, stable, efficient trusted execution environment.

Key words: trusted execution environment; trusted computing; ARM TrustZone; trusted platform module; mobile trusted module

1 引言

日益严峻的网络与系统安全问题对传统信息安全技术提出了挑战，人们开始从终端体系结构上探寻新的安全解决方案，由此产生出可信计算技术。可信计算组织（TCG, trusted computing group）推荐使用硬件和软件组件扩展普通计算平台来实现可信计算，其思路是以一个硬件安全模块（即后文的可信计算模块）为基础建立可信的计算环境，确保系统实体按照预期的行为执行[1]。TCG 于 2004 年发布了以 PC 和服务器为主要应用环境的可信平台模块（TPM, trusted platform module）标准[2]，详细规定了上述硬件安全模块的功能、软硬件接口、安全特性和实现方式。TPM 作为计算平台的信任锚，具备构建可信计算环境和远程证明所需的各类功能，通常被实现为安装在计算设备主板上的硬件芯片。

收稿日期：2014-07-02
基金项目：国家自然科学基金资助项目（91118006, 61202414）；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 (91118006, 61202414); The National Key Basic Research Program of China (973 Program) (2013CB338003)

随着移动、嵌入式和云计算等新计算环境的产生和发展，人们开始研究在新类型平台上建立可信计算环境的方法，但是由于新型平台与传统 PC 平台存在差异，直接将现有可信计算技术应用在新平台上存在许多问题。对于移动、嵌入式设备，其计算和存储能力都低于 PC，并且处理器架构也与 PC 不同，这些区别导致 TPM 不能用于移动、嵌入式环境。此外，无论是移动智能终端还是工业嵌入式设备，都需要具备较小的外形尺寸和较低的成本，并且对系统总体功耗有严格的限制，在这些设备中增加一个 TPM 芯片意味着增大体积、增加额外的硬件成本与功耗，因此 TPM 芯片的使用形式也不能被移动、嵌入式设备所接受。对于云计算环境，云数据中心采用虚拟化架构在一个服务器上运行多个虚拟机（VM, virtual machine），将 TPM 用在宿主物理服务器上能够保护客体 VM，但是与传统 PC 平台架构不同的是，虚拟化平台需要利用一个物理 TPM 芯片为多个 VM 提供信任服务，必须解决共享信任根的问题。此外，受 TPM 保护的 VM 在迁移时会由于 2 台物理服务器的硬件信任根不同，而在目标服务器上不能正常运行，因此固化的 TPM 芯片也不适用于云计算环境。

由于 TPM 芯片不能满足新计算环境的需求，因此设计新的方案来为这些环境提供可信计算功能受到广泛关注。这方面的研究已经取得一些成果，但是现有方案仍存在一些缺陷：1) 不能跨平台使用。各方案都根据其应用平台的特点设计，不适用于其他平台，从而导致不同类型终端上的可信计算环境不同，某一终端处理的数据会因为互操作问题而不能直接放到其他类型终端上使用，然而用户通常拥有 PC、智能手机、平板电脑等不同的设备，上述问题会给用户在设备间传输数据带来不便。2) 未考虑功能更新。各方案都遵循 TCG 标准的当前版本设计和实现，对标准升级后的功能和算法更新考虑较少，如果 TCG 发布新的标准，现有方案会因为不符合新标准而不能继续使用。

针对上述问题，本文基于 ARM TrustZone 技术，提出一种硬件可信执行环境模块（TEEM, trusted execution environment module）架构。其特点是：1) 能够在多种类型终端上建立可信计算环境，可以为 PC、移动/嵌入式设备、云服务器上的应用程序提供可信计算功能；2) 以模块化的方式实现 TEEM 的各种功能和算法，支持可信计算功能和算法更新；3) 通过一个可变命令集保证 TEEM 功能和算法的灵活性，使用户能根据需要对 TEEM 进行配置；4) 以移动设备作为 TEEM 的载体，用户能直接将智能手机等移动终端作为便携式可信计算设备在不同平台上使用，具有较强的移动性和便携性。

2　技术背景和相关工作

2.1　技术背景

2.1.1　可信计算技术

可信计算是一项由 TCG 推动和开发的技术，目的是在计算和通信系统中广泛使用可信计算模块支持的可信计算平台，提高系统和网络整体的安全性。TCG 定义了 TPM 和 MTM（mobile trusted module）2 种可信计算模块，MTM 标准[3]是基于 TPM v1.2 标准构建的移动可信模块标准，其命令由 TPM v1.2 中的部分命令和一些额外增加的命令组成，因此 MTM 可以看成是更适合移动平台的特殊 TPM。我国致力于发展自主创新的可信计算，推出了拥有自主知识产权的可信密码模块(TCM, trusted cryptography module)标准[4]，TCM 与 TPM 的区别在于密码算法自主化，TCM 采用了国家密码管理局指定的算法。不论是 TPM、MTM 还是 TCM，各可信计算模块标准的差异主要体现在算法和命令集上，但其核心思想相同，即为平台提供密码学功能和受保护的存储空间，完成完整性度量、存储保护、密钥管理和远程证明等功能，利用这些手段构建一个安全可信的软硬件运行环境。

TCG 标准发布后被众多软硬件厂商支持和采用，实现 TCG 硬件规范和相关软件应用的产品相继出现。近期 TCG 发布了 TPM v2.0 标准草案[5]，与当前的 TPM v1.2 标准相比增加了许多新特性，如提供灵活的密码算法和统一的授权框架。该标准获得了微软推出的 Windows 8 系统的支持，该系统基于 TPM 进行了多项安全功能扩展。随着 Windows 8 的发布 TPM 在操作系统中的重要性和作用显著增强。

2.1.2　ARM TrustZone 技术

ARM 处理器是目前应用最广泛、影响力最大的移动/嵌入式微处理器类型之一。TrustZone 技术是 ARM 公司为移动平台设计的新型 ARM 处理器安全增强技术，主要功能是构建可信的计算与 I/O 环境。该技术通过将软硬件资源划分为“安全世界”和“普通世界”，并在两者间实施严格的单向隔离：

普通世界不能随意访问安全世界，使安全世界成为一个可信的环境。

TrustZone 技术的实质是引入安全和普通两种处理模式到处理器内核，处理器内部系统控制协处理器的安全配置寄存器增设了 NS 比特位，用于指示处理器执行模式。普通模式的处理器只能访问普通世界的软硬件资源，安全模式的处理器可以访问所有资源。处理器包含一个特殊的安全监控模式和一个安全监控调用（SMC, secure monitor call）指令用于联系 2 个世界，系统级代码在监控模式下可以进行执行模式切换。在处理器体系结构增强的基础上，为保证各种 I/O 操作的可信，TrustZone 技术为系统和外围总线引入了一种类似于处理器 NS 比特位的新控制信号，使外围设备也具备区分安全与普通世界并借此进行基本访问控制的能力。

TrustZone 软件利用上述硬件安全扩展保护安全 OS 和安全外设不受运行在普通世界的代码的危害。该软件提供一个最小的安全内核，该内核与普通世界的功能更齐全的高级 OS 并行运行在同一处理器内核上，并为普通世界 OS 提供与安全世界 OS 通信的驱动程序。

2.2 相关工作

为满足不同类型平台的可信计算功能需求，国内外研究人员对 PC、移动、嵌入式和云计算环境中的可信计算环境构建方案展开了研究。

PC 平台使用 TPM 时需要将其固定在主板上并通过 LPC 总线进行连接，这种使用方式不够灵活。文献[6,7]提出便携式 TPM 即 PTPM 方案，PTPM 与 TPM 基本功能相同，但通过 USB 接口与 PC 连接，其实质是一个具有可信计算功能的高性能安全 USB Key，利用可扩展固件接口，系统能够在平台启动阶段加载 USB 驱动来识别 PTPM，使 PTPM 能作为整个平台的信任根建立信任链。

对于移动平台，文献[8]提出拆分和动态加载 MTM 的思想，并基于 M-Shield 技术实现了移动平台上的 MrTM，方案重点分析了如何减小 MrTM 的资源消耗，但是对系统安全方面的设计细节论述较少。文献[9]设计并实现了基于 Java 卡的可信计算功能体系结构，引入 SIM 卡和智能卡提供计算和存储保护，但是方案不支持安全启动和可信引擎隔离，而且只支持一个 MTM 实例和少量核心的 MTM 命令。为降低 MTM 所需的资源，文献[10]采用与文献[8]相同的思路研究出基于 Java 技术的 MTM 拆分和动态加载方案，但是方案主要侧重于提高性能，安全性分析比较粗略。文献[11]在支持 TrustZone 的移动平台中实现了可信计算功能，方案利用虚拟机架构和操作系统内部的强制访问控制机制实现了可信引擎的隔离。文献[12]提出基于 TrustZone 和.NET 技术为移动应用构建可信运行时环境，方案允许应用程序组件运行在与 OS 和其他应用隔离的可信环境中，并保护其代码和数据的完整性和机密性。文献[13]提出轻量级的 OS 级虚拟化方案 AirBag，方案动态实例化一个虚拟隔离环境运行不可信应用，防止其破坏原生系统和泄露用户信息，从而增强 Android 平台对恶意软件的防御能力，但作者指出 AirBag 特征明显，很容易被应用程序辨别并对其进行有针对性的攻击。文献[14]将移动 OS 的关键安全服务委托给一个单独的模块 μTCB，μTCB 为应用程序提供核心服务，并管理敏感数据和手机传感器，从而减轻智能手机的恶意软件攻击，作者基于 TrustZone 实现了方案的原型系统，但该原型系统的组件全部在普通世界中实现，未涉及对安全世界的访问和使用。

对于嵌入式环境，文献[15]提出基于 FPGA（field programmable gate array）平台实现可信计算功能的方案，其思路是将 TPM 和应用软件代码存储在 FPGA 外部并在运行时动态加载，方案能为应用层逻辑提供完整性保护，但是需要 FPGA 内置多次可编程非易失性存储器（MTP-NVM），而目前 FPGA 还未实现该技术。因此文献[16]又提出在只有一次可编程非易失性存储器（OTP-NVM）的 FPGA 平台上实现 TPM 功能的方案，将需要 MTP-NVM 存储的秘密信息内置于 OTP-NVM 和外部的非易失性存储器中。文献[17]提出用 PUF（physically unclonable function）产生 AES 密钥保护可信执行环境的框架，方案重点讨论了如何在 TrustZone 平台上实现安全引导，但是没有说明可信执行环境如何构建，作者指出其框架可以利用 FPGA 技术实现，但并未给出具体实现方式。

对于云计算环境，文献[18]提出虚拟 TPM 即 vTPM 方案，将软件 TPM 集成到虚拟化环境中从而为每个 VM 虚拟一个 TPM，使一个硬件平台上的多个 VM 都可以使用可信计算功能，但是 vTPM 没有真正的 EK，无法对外提供有效的证明。文献[19]提出一种半虚拟化的 TPM 共享方案，在 hypervisor 中提供对多个 VM 访问 TPM 的支持，但是增加了

hypervisor 的代码量和复杂性。文献[20]提出基于硬件的虚拟信任根方案，在 TPM 芯片中为每个 VM 提供单独的上下文，使所有 VM 可以共享 TPM，但是方案需要修改硬件 TPM 的体系结构。文献[21]提出用云技术扩展 TPM 功能的 cTPM 方案，方案在 TPM 中增加一个与云共享的根密钥，使用户可以在其拥有的多个设备上创建和共享受 TPM 保护的密钥和数据，并通过云扩展 TPM 的 NV 存储，使 TPM 获得一个大容量高性能的非易失性存储，但是方案假定云是可信的，而这一假设过于强大，作者在文中也特别指出了这一问题。

尽管目前已有许多在特定计算平台上构建可信计算环境的方案，但是现有方案仍存在一些问题。首先，各方案仅支持单一平台，不适合跨平台使用，而不同类型的终端设备要进行互操作，必须获得可信计算环境的跨平台支持；其次，为解决一些已发现的安全问题，可信计算相关标准不断更新，对可信计算功能、算法和协议进行升级，但是各实现方案只注重对当前标准的符合性，没有考虑如何进行升级以符合新的标准，特别是一些硬件实现方案，如 PTPM，由于存在纯硬件方式固有的限制而根本不具备升级的能力。此外，各方案提供可信计算功能的方式缺乏灵活性，用户不能对需要的功能进行配置，而额外的复杂功能无疑浪费了资源。因此，为解决以上问题，本文基于 TrustZone 技术提出 TEEM 架构。

3　可信执行环境模块架构概述

3.1　设计目标

针对现有可信计算环境构建方案存在的问题，本文将设计重点放在提出一种能为多种平台提供可信执行环境（TEE, trusted execution environment）并支持功能配置和升级的通用安全模块架构，从而提出以下功能和安全目标。

功能目标。

1) 为多种平台提供可信计算功能。TEEM 应能够为多种计算环境提供基本的可信计算功能，为应用程序提供密码算法和受保护的存储空间。

2) 可配置与升级。TEEM 应能够以模块化的方式提供可信计算功能，使用户可以根据需要配置可信计算功能集合，并且可以通过更新组件实现功能和算法升级。

3) 与传统软件环境兼容。使用 TEEM 不应要求重新设计当前的传统操作系统或其他运行在系统中的传统软件，TEEM 应符合已经发布的可信计算模块标准，已有的依赖于 TPM/TCM 的可信计算应用系统转换为使用 TEEM 提供的可信执行环境的代价应尽可能小。

4) 低性能开销。TEEM 应具有较低的性能开销，能用于资源受限的移动、嵌入式设备，上层应用使用可信计算功能时，应仅有较短的时间延迟。

安全目标。

1) 安全启动。TEEM 应支持可信计算环境的安全启动，不仅能够对启动序列进行度量，而且能够终止任何非预期的状态转换。

2) 存储保护。TEEM 应具有安全的存储空间用于存储机密数据和程序代码，能够保证数据和代码的机密性与完整性，并且具有用于加载状态和运行时数据的屏蔽的存储单元。

3) 隔离的执行环境。TEEM 应为访问屏蔽数据存储单元的程序代码提供隔离的执行环境，并且应用程序只能通过特定的接口与该环境交互。

3.2　应用场景

本文基于 TrustZone 技术实现 TEEM，利用这种硬件架构扩展技术，能重用移动、嵌入式设备已有的处理器资源，不会增加额外的硬件成本和功耗。总体思路是令 TEEM 运行在移动/嵌入式平台的安全世界，既可以直接为移动/嵌入式设备所用，又可以通过将设备与 PC 连接被 PC 平台使用，这样既保证 TEEM 受到基于硬件的安全隔离环境的保护，避免纯软件实现方式可能遭受的各种软件攻击，又使 TEEM 同 PC 物理分离，避免传统 TPM 与 PC 紧耦合方式带来的各种不便，如不能在多个 PC 上使用同一 TPM。

如图 1 所示，TEEM 有以下 3 种应用场景：1) 对于移动/嵌入式设备，TEEM 运行在 TrustZone 的安全世界，与运行在普通世界的移动操作系统和移动应用隔离，为整个平台提供可信计算功能，TrustZone 安全扩展保护 TEEM 的代码和数据不受普通世界组件的干扰和破坏；2) 对于 PC 和服务器，TEEM 设备驱动库运行在应用层，主机通过 USB 接口与带有 TEEM 的移动设备进行连接，将该设备作为便捷式可信平台模块外设使用，主机上的安全应用通过 TEEM 设备驱动库调用 TEEM 功能；3) 对于云计算环境中的服务器，TEEM 运行在隔离驱动

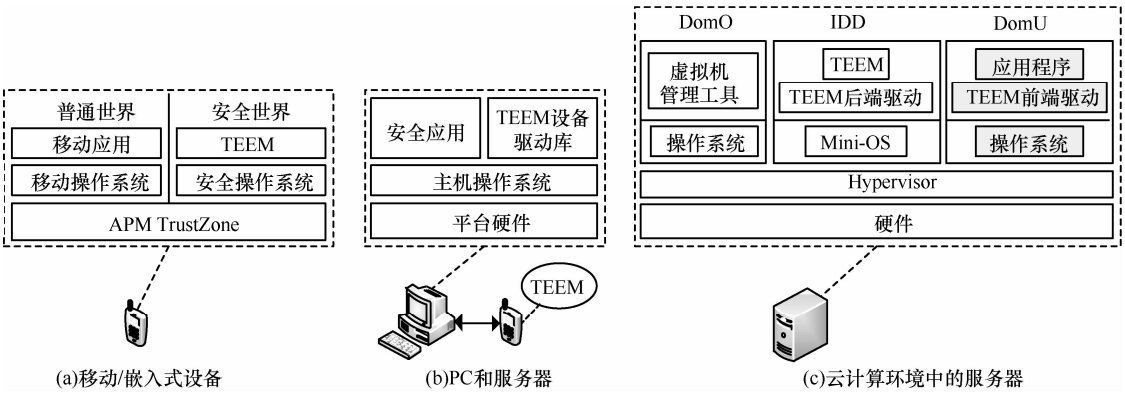

图 1　TEEM 的应用场景

域（IDD, isolated driver domain）中，受 hypervisor 隔离机制的保护，普通虚拟域（DomU）中的应用程序通过 TEEM 前端驱动向 IDD 中的 TEEM 后端驱动发出可信计算功能请求调用 TEEM 功能。

3.3　体系结构

3.3.1　逻辑结构

TEEM 架构的逻辑结构如图 2 所示，在支持 TrustZone 技术的 ARM 平台上，运行环境被隔离为 2 个区域：安全世界和运行移动操作系统、大部分移动应用的普通世界。安全世界的软件代码和运行状态的完整性和机密性受 TrustZone 的保护，不会被普通世界的不可信代码察觉或修改，因此对普通世界的软件而言，安全世界的软件可以被看作是硬件信任根。

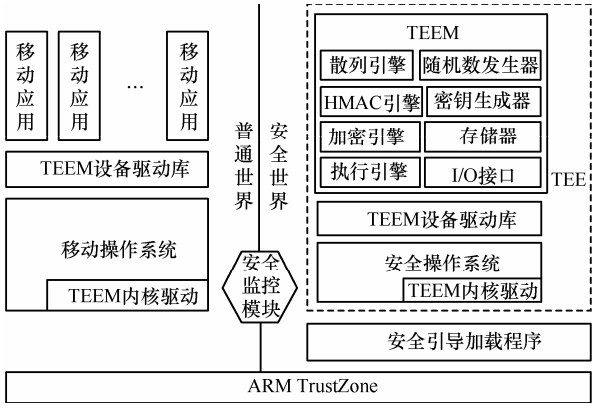

图 2　TEEM 架构逻辑结构

TEEM 是一个具有密码运算能力和存储功能的组件，包含执行引擎、加密引擎、I/O 接口、存储器、随机数发生器等部件，兼容 TPM、TCM、MTM 标准，能够完成安全存储、度量和报告、身份认证等安全功能，是建立 TEE 的基础。安全操作系统中包含 TEEM 内核驱动，是 TEE 的核心软件层。位于应用层的 TEEM 设备驱动库是调用 TEEM 功能的接口，它与 TEEM、安全操作系统共同构成了 TEEM 架构的 TEE。此外，安全世界还提供一个安全引导加载程序，用于支持 TEEM 宿主平台（简称 TEEM 平台）的安全启动。

应用程序调用 TEEM 功能时，平台需要将处理器执行模式切换到安全世界来处理调用请求。TEEM 架构通过一个安全过程调用（SPC, secure procedure call）机制提供 2 个运行环境间的安全通信信道，使被隔离的 2 个区域能够交互，该机制通过图 2 中的 4 个组件实现：应用层的程序库—普通世界 TEEM 设备驱动库和安全世界 TEEM 设备驱动库；操作系统层的通信驱动程序—普通世界 TEEM 内核驱动和安全世界 TEEM 内核驱动。操作系统层的通讯驱动程序负责实现 TrustZone 环境切换，应用层的程序库则负责将输入输出数据交付给通讯驱动程序。

3.3.2　软件结构

TEEM 架构的软件结构分为 PC 平台和移动平台 2 部分，如图 3 所示。PC 平台上的软件用于调用 TEEM，而移动平台上的软件则用于实现 TEEM 功能和与 PC 进行通信，整个移动平台都受到安全世界分区和安全操作系统内核的控制。下面按功能分别介绍各组件。

1) 安全引导加载程序：用于实现 TEEM 平台的安全启动，负责验证安全世界操作系统内核完整性，只有内核度量值与预期值一致时，才会将控制权移交给安全世界操作系统内核。

2) TEEM 组件：用于提供 TEEM 功能。TEEM 守护进程是 TEEM 的通信模块，负责监听和解析 TEEM 命令，调用可信计算功能模块进行处理，并返回得到的响应；可信计算功能模块负责处理

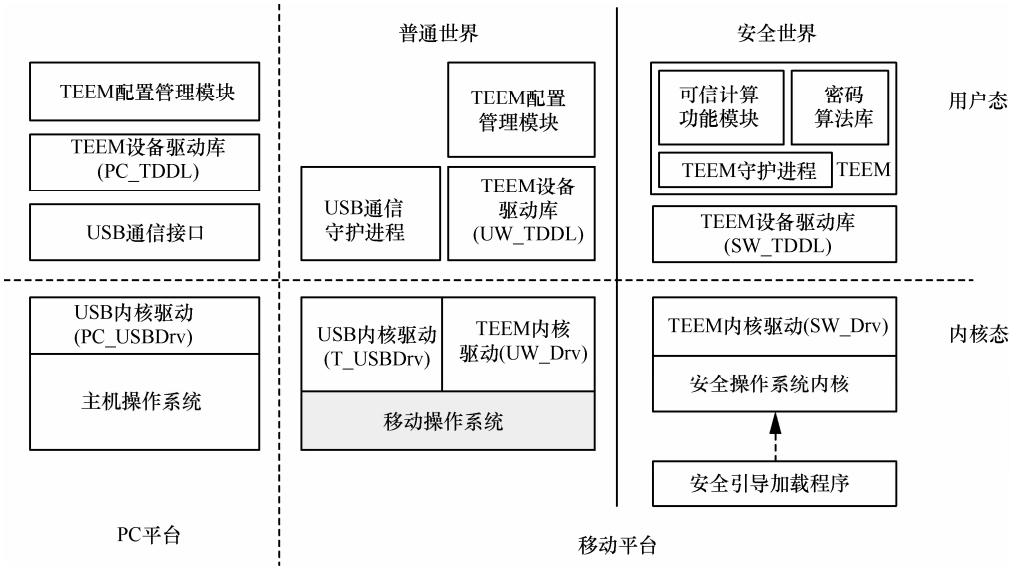

图 3　TEEM 架构软件结构

TEEM 命令，包含不同的可信引擎，如 TPM、TCM、MTM，并可以根据需要扩展新的功能；密码算法库提供 TEEM 所需的密码算法，供可信计算功能模块调用，同样也支持密码算法扩展。

3) SPC 组件：即 3.3.1 节中用于实现 SPC 机制的 4 个组件。UW_TDDL 是普通世界访问 TEEM 的标准化接口，负责发送 TEEM 命令给 UW_Drv 和接收响应；UW_Drv 通过发出 SMC 指令实现环境切换，向安全世界中的 SW_Drv 发送 TEEM 命令并接收响应；SW_Drv 将来自普通世界的 TEEM 命令发送给 SW_TDDL 处理，并通过环境切换将响应返回给普通世界中的 UW_Drv；SW_TDDL 是安全世界访问 TEEM 的常规接口，用于与 TEEM 连接，发送命令请求和接收相应的响应。

4) USB 通信组件：实现 PC 和 TEEM 平台间 USB 通信的组件。USB 通信接口是 PC 上应用程序访问 PC_USBDrv 的接口；PC_USBDrv 是标准 USB 驱动程序，能识别 TEEM 设备上的 T_USBDrv 并通过 USB 连接线与其通信；T_USBDrv 是 USB 通信驱动程序，控制 USB 设备功能的实现，从而将 TEEM 所在的移动设备呈现为一个可信平台模块外设供 PC 使用，该组件将来自 PC 的命令请求发送给 USB 通信守护进程，并返回收到的响应给 PC；USB 通信守护进程是 USB 通信模块，负责监听来自 T_USBDrv 的请求，调用 UW_TDDL 将请求发送给 TEEM 组件进行处理，并向 T_USBDrv 写入收到的响应。

5) TEEM 配置管理模块：2 个平台上的该组件功能相同，都是提供给用户的图形化界面，用于设置 TEEM 的参数、查询相关状态信息、配置需要的可信引擎和密码算法等。

6) TEEM 设备驱动库（PC_TDDL）：PC 平台上应用程序调用 TEEM 功能的常规接口，被封装为符合 TDDL 标准[22]的形式供应用程序使用。

上述组件中，TEEM 组件使 TEEM 能灵活的配置和更新，保证了 TEEM 架构的实用性。配置 TEEM 功能通过选择可信计算功能模块中的不同命令集合实现，而更新可信计算功能和协议则通过升级可信计算功能模块实现。此外，还可以通过升级密码算法库更新 TEEM 的密码算法。

3.4　基本功能

TEEM 主要具有以下 4 个基本功能。

1) 身份标识：TEEM 初始化阶段生成一个唯一的模块身份，该身份是一个非对称密钥对，用于认证 TEEM。身份密钥由用户口令加密后安全存储在 TEEM 内部，使用该密钥需要获得用户授权，用户输入口令后才能解密身份密钥，这种使用方式实质上是将 TEEM 身份与用户身份绑定。此外，只有 TEEM 的可信计算功能模块能访问身份密钥私钥，该私钥不会透露给任何其他程序或出现在安全世界外部。

2) 完整性存储和报告：TEEM 内部平台配置寄存器（PCR, platform configuration register）依赖安全世界的存储器实现，位于受保护的存储空间，应

用程序能将完整性度量结果安全存储到 TEEM 中。TEEM 平台启动时，PCR 会被重置为默认值，平台启动后，只有 TEEM 的可信计算功能模块能够通过扩展更改 PCR 值。应用程序可以使用 TEEM 引证功能进行完整性报告，在报告过程中 TEEM 签名 PCR 值并输出签名结果。

3) 数据保护：TEEM 的加密与封装功能为用户提供敏感数据的保护存储。加密和解密是最基本的数据存储功能，TEEM 能够提供非对称和对称加解密功能。此外，TEEM 的数据封装功能可以将机密数据与 TEEM 所在平台的某种平台配置绑定，被封装的数据只有在使用该 TEEM 的平台处于特定配置时才能被解封。

4) 功能配置：通过 TEEM 配置管理模块，用户能以不同粒度灵活配置 TEEM 的功能：在较粗粒度，可直接指定将 TEEM 作为 TPM、TCM 或 MTM 使用；在较细粒度，可以设定 TEEM 提供特定的可信计算功能和密码算法。

4 关键技术

4.1 平台安全启动

安全启动过程是移动可信计算的重要安全机制，是移动/嵌入式设备上建立可信计算环境的基础。TEEM 平台没有用于安全启动的硬件 MTM，而软件 MTM 在安全世界 OS 内核调用用户空间的 TEEM 程序时才可用。基于上述原因，TEEM 平台的安全启动过程需要依靠平台 SOC ROM（system on chip read only memory）中的安全引导加载程序。该程序的任务是在移交控制权给安全世界 OS 内核之前认证内核映像和参数：程序度量安全世界 OS 内核映像，验证参考完整性度量（RIM, reference integrity metric）证书中的值是否与获得的度量值一致，只有验证成功，引导过程才能继续，程序才会将系统控制权和度量值交给安全世界 OS 内核，并且内核转移控制权给用户空间的应用程序后该度量值仍然可用。图 4 概述了 TEEM 平台的安全引导流程。

在 TEEM 平台上，任何安全世界的程序映像和内核模块都必须携带 RIM 证书，安全世界 OS 内核支持一个与安全引导加载程序类似的机制，用于验证用户空间的应用程序和待加载的内核模块。安全世界 OS 内核首先验证并加载 TEEM，之后 TEEM 就可以被用于 TCG 方式的 RIM 证书验证和扩展[3]。内核加载其他程序映像或内核模块时会度量其完整性并调用 TEEM 验证 RIM 证书，如果验证失败，就拒绝加载违规的映像或模

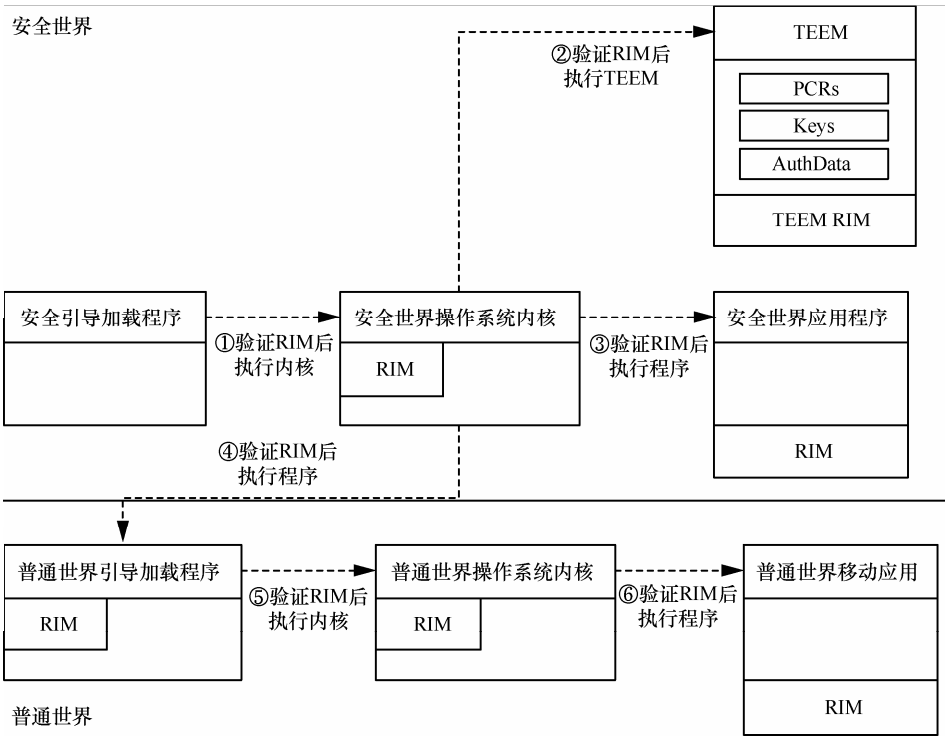

图 4 TEEM 平台安全引导流程

块。安全世界的应用程序加载结束后，内核度量普通世界的引导加载程序，调用 TEEM 验证其 RIM 证书，验证成功后将控制权交给该程序，开始进行普通世界启动过程。

普通世界 OS 内核和应用程序的完整性验证都通过 MTM_VerifyRIMCertAndExtend 命令调用安全世界中的 TEEM 完成。首先普通世界引导加载程序度量普通世界 OS 内核映像，并调用 TEEM 验证内核 RIM 证书，验证成功后，将控制权交给该内核。然后内核采用类似的方式度量、验证普通世界的应用程序和内核模块，确保只有验证成功的程序和模块才能被加载。当用户空间的应用程序都成功执行时，就完成了 TEEM 平台的安全启动过程。

4.2　安全过程调用

由于实施了安全世界和普通世界间应用程序的隔离，TEEM 平台需要建立 2 个世界间安全的通信信道才能使不同世界的应用程序跨越安全边界进行通信，为此 TEEM 架构提供了 SPC 机制。如 3.3.2 节所述，SPC 机制由图 3 中的 4 个组件实现，其中 UW_Drv 和 SW_Drv 这 2 个内核驱动模块是 SPC 机制的核心组件，它们通过使用 TrustZone 指令实现安全模式和普通模式的切换，并负责编码参数和返回值。一个 SPC 请求过程可以归纳为以下 4 步。

1) UW_TDDL 接收普通世界应用程序发送的 TEEM 命令后，产生 SPC 请求发送给 UW_Drv；

2) UW_Drv 执行 ARM 处理器提供的 SMC 指令产生处理器异常，导致处理器进入安全监控模式并跳转到 SW_Drv 实现的异常处理程序；

3) SW_Drv 通过保存普通世界处理器状态并恢复安全世界处理器状态实现环境切换，之后处理器离开安全监控模式，SW_Drv 将请求转发到 SW_TDDL；

4) SW_TDDL 调用 TEEM 响应 SPC 请求。

TEEM 处理完成后，系统采用相同的机制沿相反的路径返回普通世界。

4.3　命令处理流程

下面以 PC 上的应用程序调用 TPM_Sign 命令为例，使用图 3 中的组件名称介绍 TEEM 的命令处理流程。首先用户通过 TEEM 配置管理模块设置 TEEM 启动 TPM 可信引擎，然后应用程序就可以使用 TEEM 中的 TPM 功能。命令处理流程如图 5 所示，主要包括以下 8 个步骤。

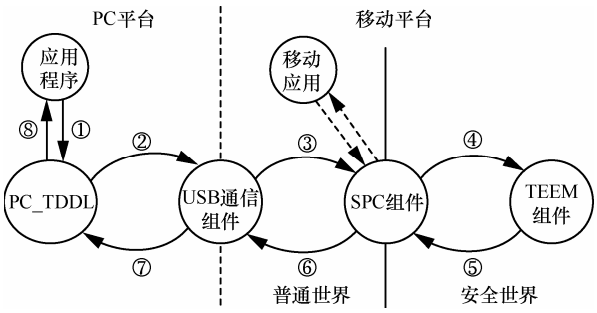

图 5　TEEM 命令处理流程

① 应用程序调用 PC_TDDL 的 Tddli_TransmitData()接口发送 TPM_Sign 命令，发起 TEEM 命令请求；

② PC_TDDL 利用 USB 通信组件将命令请求发送到 TEEM 平台：

a) PC_TDDL 通过 USB 通信接口提供的 I/O 接口，将命令请求发送给 PC 上的 USB 内核驱动 PC_USBDrv；

b) PC_USBDrv 通过 USB 连接线将命令请求进一步发送给 TEEM 平台的 USB 内核驱动 T_USBDrv；

c) T_USBDrv 收到来自 PC 的命令请求后，将其转发给用户空间的 USB 通信守护进程进行处理；

③ USB 通信组件调用 SPC 组件将命令请求发送到安全世界：

a) USB 通信守护进程监听到 T_USBDrv 的命令请求后，调用普通世界的 UW_TDDL；

b) UW_TDDL 将命令请求通过 SPC 机制发送给安全世界的 SW_TDDL；

④ SPC 组件调用 TEEM 组件处理命令请求，处理过程如下：

a) TEEM 守护进程监听到 SW_TDDL 发送的命令请求后，将其发送给正在运行的 TPM 可信引擎处理；

b) TPM 可信引擎解析请求数据，根据命令码调用相应的功能模块处理该命令；

c) TPM 可信引擎编码得到的处理结果，生成命令响应返回给 TEEM 守护进程；

⑤ TEEM 组件生成的命令响应通过 SPC 组件发送到普通世界；

⑥ SPC 组件调用 USB 通信组件将命令响应发送到 PC 平台；

⑦ USB 通信组件将命令响应返回给 PC_TDDL；

⑧ PC_TDDL 将命令响应返回给发起该命令

请求的应用程序。

移动平台上的应用程序调用 TEEM 命令的流程则相对简单，移动应用通过调用 SPC 组件将命令请求发送到安全世界，然后经过上述④、⑤ 2 个步骤后，SPC 组件就可以将命令响应返回给发出该命令请求的移动应用。

4.4 实施方案

本节详细说明 TEEM 如何在移动/嵌入式设备、PC 上使用，包括设备软硬件基础、TEEM 的配置与使用方法和平台安全启动等，并简要说明 TEEM 在云环境中的应用。

4.4.1 移动/嵌入式设备

移动/嵌入式设备使用TEEM的总体结构如图6所示，要求设备的微处理器支持 TrustZone 技术，并将软件安装在正确的隔离区域。除了位于安全世界的 TEEM 外，移动/嵌入式设备若要使用 TEEM 必须安装 SPC 组件和 TEEM 配置管理模块。此外，设备还可以安装一些可选的组件：TEEM 可信软件栈（TSS, trust software stack），该软件为应用程序提供使用 TEEM 功能的统一 API 接口，隐藏构建命令流的底层细节，使访问 TEEM 更加简单和直接；USB 通信组件，即 USB 通信守护进程和 USB 内核驱动，用于实现 TEEM 设备与 PC 的 USB 通信，如果用户不仅要在移动设备上使用 TEEM，还要在 PC 上使用，那么就需要安装该组件。

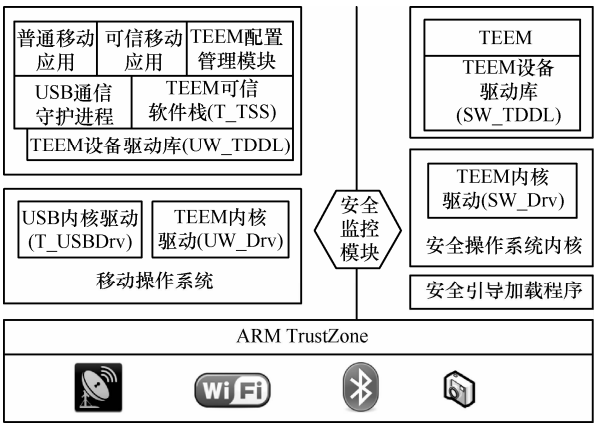

图 6　移动设备使用 TEEM 的总体结构

软件安装完成后，用户首先应通过 TEEM 配置管理模块配置可信引擎，可以直接将其设置为 MTM，或者在 MTM 功能的基础上进一步配置和裁剪所需的功能。配置好 TEEM 后，用户需要将安全引导加载程序置于移动设备的 SOC ROM 中，用于认证安全世界操作系统内核启动前平台环境的可信性，之后就可以重新启动设备。设备启动后首先进行安全引导过程，实现整个移动平台的安全启动，启动完成后 TEEM 就可以正常使用。可信移动应用可以通过 UW_TDDL 或 T_TSS 调用 TEEM 功能，而底层 2 个隔离区域利用 SPC 机制进行通信的过程对应用程序则是完全透明的。

4.4.2 PC 平台

PC 平台使用 TEEM 的总体结构如图 7 所示，必须的软件包括内核层的 USB 内核驱动和应用层的 USB 通信接口、TEEM 设备驱动库、TEEM 配置管理模块。此外，PC 平台同样可以安装 TEEM 可信软件栈来简化 TEEM 的使用。

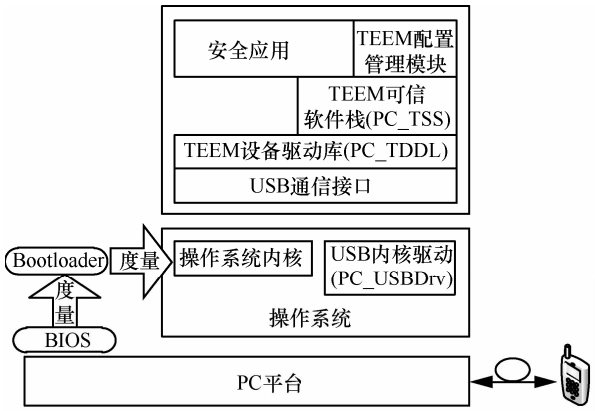

图 7　PC 平台使用 TEEM 的总体结构

安装了上述软件后，用户首先应启动带有 TEEM 的移动设备（简称 TEEM 设备），并配置需要的可信引擎，如 TPM 或 TCM，然后用 USB 连接线将设备与 PC 连接。连接完成后，PC_USBDrv 能够将 TEEM 设备识别为可信平台模块外设，并通过 USB 连接线与 TEEM 通信。此时 PC 上的应用程序就可以使用 TEEM 的功能，应用程序可以直接调用 PC_TDDL 的接口发送 TEEM 命令请求，或者通过 TSS 调用 TEEM 功能。用户也可以通过 PC 上的 TEEM 配置管理模块进一步配置 TEEM 的功能和算法。

此外，PC 平台可以利用 TEEM 实现可信引导，建立系统启动过程的信任链。在 PC 启动之前，先将 TEEM 设备与 PC 连接，然后再为系统加电，之后就可以从 PC 启动开始，以 TEEM 为信任根，在可信 BIOS 的基础上，分别对 Bootloader 和操作系统内核进行度量和验证，实现操作系统启动过程的信任链构建。引导过程中，每一阶段的度量结果都能够扩展到 TEEM 中，并且只有度量值与预期值一致时才将系统控制权移交给下一阶段要执行的程

序，最终通过整个度量和验证过程确保 PC 平台引导的操作系统是可信的。

4.4.3　云计算环境

在云计算环境中，云服务端可以将 TEEM 放到云数据中心的虚拟架构中使用，由于缺少 TrustZone 技术的支持，因此需要采用隔离的虚拟机驱动域运行 TEEM 组件以保证其安全性。利用 hypervisor 的安全隔离机制，TEEM 能够与不可信 OS、应用程序隔离，保证其数据和运行状态的完整性和机密性不受危害。TEEM 适用于云计算环境的一个重要原因在于，同一物理服务器可以运行多个 TEEM 实例，其上的 VM 可以由不同的 TEEM 实例保护，这就解决了多个 VM 共享信任根的问题。此外，TEEM 可以随 VM 迁移到其他物理服务器上，保证 VM 在目标服务器上仍然可以正常运行，解决了受保护的 VM 的迁移问题。所以将 TEEM 置于隔离驱动域后，服务器上的 VM 就可以像使用 TPM/TCM 芯片那样使用 TEEM 提供的可信计算功能，实现虚拟机证明等安全应用。

5　实验

TEEM 架构的原型系统分为移动/嵌入式平台和 PC 平台 2 部分，分别部署在 ARM 开发板和 PC 上。本节首先介绍 TEEM 原型系统，然后说明系统的通信效率和处理性能的测试结果，最后给出系统的安全性和性能评价。

5.1　原型系统

原型系统如图 8 所示，分为 ARM 开发板上的 TEEM 原型和 PC 使用 TEEM 设备所需的软件 2 部分。本节首先分别介绍这 2 部分的技术细节，然后说明 TEEM 设备与 PC 间 USB 通信的实现方式。

5.1.1　TEEM 原型

TEEM 原型运行在支持 TrustZone 技术的 ARM 开发板上，下面详细介绍 TEEM 原型中除 USB 通信组件外其他各组件的实现方法。

1) TEEM 组件。TEEM 组件以开源的 TPM Emulator 项目[23]为基础实现，在该项目的 TPM/MTM 模拟器上扩展了 TCM 功能，扩展形成的 TEEM 组件是一个 Unix 守护进程，由监听模块、命令解析模块、可信引擎和密码算法库组成，能提供丰富的可信计算功能。对 TPM/MTM 模拟器的修改主要包括以下 3 部分。

a) 添加 TCM 特定的数据结构到模拟器中，并开发用于编码和解码新增加的数据结构的实用函数；

b) 增加提供 TCM 功能的命令处理程序，并修改命令解析模块使其能解析 TCM 命令；

c) 扩展密码算法库使 TEEM 支持更多的密码算法，如 TCM 所采用的 SM2、SMS4 和 SM3 算法。此外，为增强 TEEM 的可配置性和标准兼容性，还在密码算法库中增加了基于双线性映射的密码算法[24]。

2) TEEM TDDL。基于 TPM emulator 项目中的 TDDL 实现，是访问 TEEM 的通用接口。

3) libTEEM。libTEEM 基于 IBM 软件 TPM 项目[25]中的 libtpm 库实现。libtpm 提供用于控制 TPM 和发送命令的低级别 API，通过在其上增加调用 TCM 功能的函数库和使用 TCM 功能的命令行工具扩展形成 libTEEM。libTEEM 通过 TCP/IP 套接字与 TEEM 连接，为应用程序提供使用 TEEM 功能的简单接口。

4) TEEM 管理界面。TEEM 管理界面使用 Qt 库[26]实现，为用户提供 TEEM 的综合查询、配置和测试功能。通过该界面用户可以实时了解 TEEM 的

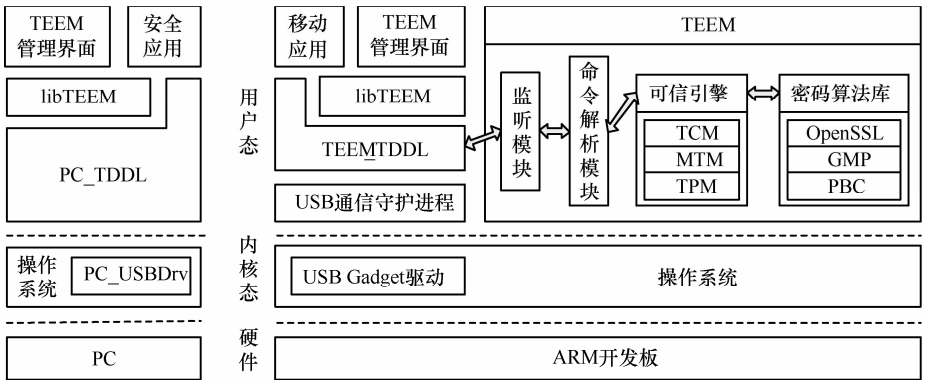

图 8　原型系统体系结构

内部信息，精确配置其运行状态和参数，以及全面测试 TEEM 的核心功能。图 9 是 TEEM 管理界面测试面板中的 TCM 解密功能界面。

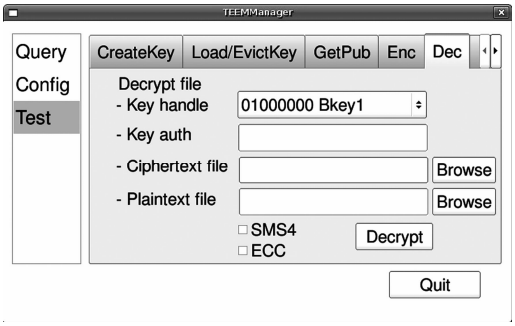

图 9 TCM 解密功能界面

5.1.2 PC 平台软件

PC 平台使用 TEEM 设备必需的 2 个软件是 PC_USBDrv 和 PC_TDDL，前者为 PC 提供与 TEEM 设备通信的手段，将在下一节详细说明，后者与 TEEM 原型中的 TEEM_TDDL 类似，只是将 I/O 接口改变为使用 USB 通信机制。此外，TEEM 原型中的 libTEEM 库和 TEEM 管理界面经过重新编译后在 PC 平台上使用，从而给应用程序提供更方便的调用 TEEM 功能的接口，并且使用户可以在 PC 端配置 TEEM。

5.1.3 USB 通信组件

2 个平台上的 USB 驱动和相应的用户空间进程使 PC 能够与 TEEM 设备进行 USB 通信。

TEEM 设备和 PC 平台上的 USB 驱动分别为 gadget 驱动和 CDC/ACM 驱动[27]，2 个驱动通过 USB 连接线通信。在 TEEM 原型的 Linux 系统中，gadget 驱动表现为一个串口驱动；而在 PC 主机系统中，gadget 驱动表现为一个兼容 CDC/ACM 类的设备，有批量输入和输出端点，被当作普通串行设备处理。Gadget 驱动已经过 Linux ACM 驱动和 Windows usbser.sys ACM 驱动的测试，确保其在不同主机操作系统中都能正常通信。在 PC 上，CDC/ACM 驱动产生一个串行端口（Linux 为 /dev/ttyACM0，Windows 为 COM），该端口可以被当作 TEEM 字符设备使用。利用上述驱动 PC 可以将 TEEM 设备识别为一个提供可信计算功能的硬件模块，因此对 PC 而言，TEEM 设备就是通过 USB 连接线与主机相连的便携式可信平台模块。

除 USB 驱动外，PC 与 TEEM 设备进行 USB 通信还需要相应的分别用于 TEEM 原型和 PC 主机的用户空间代码：USB 通信守护进程和 PC_TDDL。USB 通信守护进程监听 gadget 串口驱动（字符设备为 /dev/ttyGS0 ）的 TEEM 功能请求，调用 TEEM_TDDL 进行处理，将响应返回给 gadget 串口驱动，并最终通过 CDC/ACM 驱动返回给 PC 上的调用程序。PC_TDDL 是访问 TEEM 设备的常规接口，该库将应用程序的 TEEM 功能请求发送给 TEEM 字符设备，并返回接收到的响应。

5.2 系统测试

5.2.1 系统部署

TEEM 原型和 PC 平台软件分别部署在 Real210 开发板和联想台式机上，2 个平台的配置如下。

Real210 开发板。开发板采用三星推出的 S5PV210 应用处理器，该处理器基于 ARM Cortex-A8 架构，实现了 TrustZone 扩展，主频为 1 GHz。开发板配备有 512 MB DDR2 内存，256 MB NAND FLASH，并提供 USB 串行端口，能通过 MiniUSB 接口与主机连接。为启用开发板的 gadget 串口驱动程序，在交叉编译内核时需配置嵌入式内核（Linux-2.6.35）支持 S3C HS USB OTG Device 和 Serial Gadget，编译后产生内核模块 g_serial.ko，该模块自动加载底层 gadget 外设控制器驱动程序，通过 insmod 指令可以将该模块加载到开发板上。

PC 主机。主机配置为 2.4 GHz Intel 双核 CPU，3 GB 内存和通用 USB 接口，操作系统为 Windows XP SP3。为测试不同主机系统与 TEEM 设备间的通信效率和调用 TEEM 命令的响应时间，该主机通过 VMware 虚拟机运行一个 Ubuntu Linux 系统，该系统分配了单核 CPU 和 512 MB 内存。通过开启 Windows 系统的 VMware USB Arbitration Service 服务，并在虚拟机的可移动设备选项中勾选 TEEM 设备，可以使 TEEM 设备作为 Linux 系统的 USB 设备。

5.2.2 性能测试

测试 1 Windows 和 Linux 系统环境下，PC 平台和 TEEM 设备间 USB 通信效率。

PC 与 TEEM 设备间纯 USB 通信的性能采用以下方式进行测试：主机发送一个字节流数据给 TEEM 设备，TEEM 设备不做任何处理，直接返回一定大小的字节流数据，两者间的数据传输格式为数据长度加数据内容。TEEM 命令请求和响应数据最少为 10 byte（如 reset 命令），最多不超过 4 096 byte，本文测试了数据量在这一区间时的 USB 通信性能，但是大部分命令的传输数据量在 10 byte

到 800 byte 之间，该区间的测试结果如图 10 所示。在 Windows 主机中数据量每增加 100 byte，传输时间增加约 3 ms；在 Linux 主机中，800 byte 以内的数据传输时间差别较小，通常在 5 ms 到 10 ms 之间。总体来看，Windows 主机与 TEEM 设备间的 USB 通信时间比 Linux 主机长，而且随数据量增长的趋势明显，这与 2 个操作系统中 USB 驱动和内核模块不同有关。

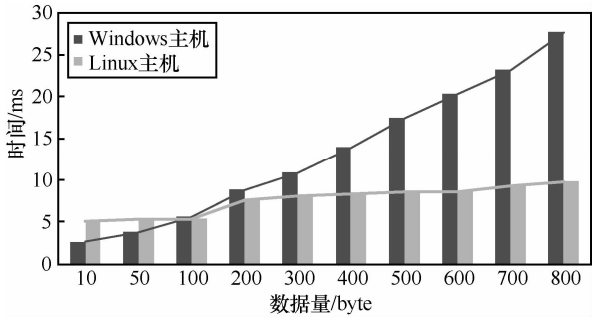

图 10　PC 主机与 TEEM 间 USB 通信时间

测试 2　TEEM 设备、Windows 主机和 Linux 主机调用 TEEM 命令的响应时间。

利用 libTEEM 库分别测试 TEEM 设备、Windows 主机和 Linux 主机上调用 TEEM 命令的响应时间，其结果如表 1 所示。测试内容包括密钥操作、密码学操作、随机数与散列操作、身份与完整性操作。在测试过程中，有些命令的请求数据中包含其他命令的响应数据，因此在调用这些命令前需要调用其他命令，如 CreateWrapKey 命令执行前需要调用创建授权会话命令，表 1 中的响应时间是调用相关命令和被测命令的总响应时间。测试结果显示，对于本身比较耗时的命令，如 CreateWrapKey，PC 与 TEEM 设备间的 USB 通信对其响应时间影响较小；而对于本身执行时间较短的命令，其响应时间则更多地受到 USB 通信的影响。此外，相同的命令在 Windows 主机上调用比 Linux 主机更耗时，这与 USB 通信的测试结果一致。

测试 3　TEEM 与 TPM、TCM 芯片执行可信计算命令的性能比较。

本测试将 TEEM 与目前国内外常用的 2 种可信计算模块：TPM 芯片、TCM 芯片进行比较，详细对比常用可信计算命令在 TEEM 和 TPM 芯片、TCM 芯片上执行时的性能差异。参与比较的 TPM 芯片为嵌入在 IBM ThinkCentre M52 8114 台式机上的 TPM v1.2 芯片，芯片上的非对称密码算法为 2 048 bit RSA 算法，主机配置为 3.20 GHz CPU，512 MB 内存，TCM 芯片为嵌入在联想 ThinkCentre M4000t 台式机上的 ZTEIC TCM v1.1 芯片，芯片上的非对称密码算法为 256 bit SM2 算法，主机配置为 2.80 GHz Intel 双核 CPU，2 GB 内存。本测试通过 libTEEM 调用 TEEM 测试其 2 048 bit RSA、256 bit SM2 算法的响应时间，并通过 TSS、TSM 分别调用 TPM、TCM 测试相同算法的响应时间，结果如图 11 所示，整体看来 TEEM 的处理性能优于上述硬件芯片。

5.3　系统评价

安全性分析　TEEM 组件运行在移动/嵌入式平台的安全世界中，受到 TrustZone 安全隔离技术的保护，其安全性远高于软件实现的可信计算模块。考虑具有以下攻击能力的敌手：敌手控制 TEEM 平台普通世界中的操作系统，可以利用操作系统和应用程序实施各种软件攻击，能访问操作系统管理

表 1　**TEEM 设备、Windows 主机和 Linux 主机调用 TEEM 命令的响应时间**

命令	TEEM/ms	Windows/ms	Linux/ms	命令	TEEM/ms	Windows/ms	Linux/ms
ReadPubek	31.8	187	55.1	StirRandom	1.9	312	330
CreateWrapKey	4432	4406	3928	SHA1Start	3.4	46.8	19.6
LoadKey	611	655	683.9	SHA1Update	3.1	31.2	19.4
EvictKey	1.9	62.5	114.5	SHA1Complete	0.8	31.2	19.4
GetPubKey	5.7	250	458	SHA1CompleteExtend	0.9	31.2	20.1
Sign	83	343	217	MakeIdentity	3240	3593	4337
UnBind	84	375	167	ActivateIdentity	111	421	526
Seal	11	288	116	PcrRead	3.3	62.5	14.2
UnSeal	89	453	169	PcrExtend	3.2	62.5	15.7
GetRandom	3.9	78	48.5	Quote	86	359	167

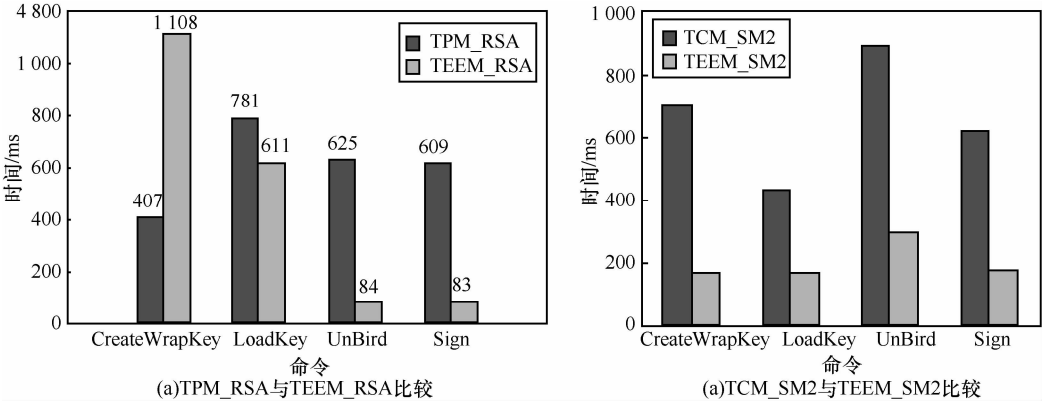

图 11 TEEM 与硬件芯片命令执行效率比较

的任何系统资源和安全服务；但是敌手不能对 TEEM 架构依赖的硬件实施物理攻击，如攻击支持 TrustZone 技术的移动/嵌入式微处理器。首先，在 TEEM 平台启动之前，如果敌手利用恶意软件篡改了 TEEM 组件的代码或配置，那么在 TEEM 平台引导时，平台安全启动过程能够检测到这种攻击，被篡改的 TEEM 组件不能通过 RIM 证书验证，平台启动过程就会终止，以防止系统进入不安全的状态，因此在 TEEM 平台成功启动前，敌手对 TEEM 组件的任何修改都能被检测出来。其次，在 TEEM 平台运行时，如果敌手试图利用恶意软件篡改 TEEM 组件、干扰 TEEM 执行或访问 TEEM 保护的数据，这种类型的攻击都会被 TrustZone 技术防止，TrustZone 的硬件隔离机制可以保护 TEEM 组件的代码和数据内存不受普通世界代码的危害，因此在 TEEM 平台运行过程中，敌手不能成功实施对 TEEM 组件的攻击。综合以上分析可见，TEEM 架构能够防止来自敌手的软件攻击，可以保证 TEEM 平台启动和运行时 TEEM 组件的安全性和执行完整性，以及受 TEEM 组件保护的数据的机密性和完整性。

性能分析 下面根据原型系统的测试结果分析 TEEM 架构的性能。首先，在 TEEM 设备端，Real210 开发板上 TEEM 命令的执行性能是高效的，大部分可信计算功能，如 Quote 操作，仅需要不足 90 ms 的时间。其次，在 PC 平台端，TEEM 命令响应时间包含 Real210 开发板上执行 TEEM 命令的时间和主机与 TEEM 设备进行 USB 通信的时间。实验结果表明：对于数据传输量大的命令，其通信代价较之命令执行时间可以忽略，但是对于数据传输量较小的命令，其命令响应时间会明显受到 USB 通信时间的影响。虽然 PC 平台上 TEEM 命令响应时间会受 USB 通信的影响，但是大部分命令的响应时间都低于 460 ms，延迟时间很短。最后，在性能对比实验中，TEEM 创建 RSA 密钥的响应时间为 1 108 ms，慢于 TPM 执行该命令的时间，原因在于 TPM 在处理该命令时进行了优化，而本文的原型系统目前并未进行该优化，但是综合来看，TEEM 的命令执行效率要高于硬件芯片，并且原型系统还可以通过优化进一步提高效率。通过上述分析可以看出，TEEM 架构能够高效的提供可信计算功能，无论在 TEEM 设备还是 PC 上，都能快速响应 TEEM 命令请求，并且其命令处理性能高于 TPM/TCM 芯片。

6 结束语

本文提出一种用硬件实现通用可信执行环境模块的方案，该方案利用 ARM TrustZone 技术提供的硬件隔离机制保护 TEEM 组件的安全性，能够为多种计算环境提供可信计算功能。TEEM 可以应用于传统 PC 环境，并且可以在不明显降低 TEEM 安全性的前提下，重点应用于移动智能终端、工业嵌入式设备和云计算环境中的服务器节点。对于移动/嵌入式设备、PC 平台，TEEM 架构可以完全支持，前者可以将 TEEM 组件放到 TrustZone 的安全世界中使用其功能，设备本身也就成为了一个 TEEM 设备，后者通过 USB 连接线与 TEEM 设备连接后，就可以像使用 TPM/TCM 芯片一样使用 TEEM。对于云计算环境，由于服务器缺少 TrustZone 技术的支持，因此需要采用隔离的虚拟机驱动域运行 TEEM 组件以保证其安全性。为验证方案的可行性，本文实现了 TEEM 架构的原型系统，原型系统的测试结果表明：TEEM 是为不同平台提供可信执行环境的高效方案，并且处理性能优于硬件可信安全芯片。

参考文献：

[1] 冯登国, 秦宇, 汪丹等. 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, WANG D, *et al*. Research on trusted computing technology[J]. Journal of Computer Research and Development, 2011, 48(8): 1332-1349.

[2] Trusted Computing Group. TPM main specification version 1.2[EB/OL]. http://www.trustedcomputinggroup.org, 2014.

[3] Trusted Computing Group - Mobile Phone Work Group. TCG mobile trusted module specification version 1.0[EB/OL]. http://www. trustedcomputinggroup.org, 2014.

[4] 国家密码管理局. 可信计算密码支撑平台功能与接口规范[EB/OL]. http://www.oscca.gov.cn/UpFile/File64.PDF, 2014.
State Cryptography Administration. Functionality and interface specification of cryptographic support platform for trusted computing[EB/OL]. http://www.oscca.gov.cn/UpFile/File64.PDF,2014.

[5] Trusted Computing Group. Trusted platform module library[EB/OL]. http://www.trustedcomputinggroup.org, 2014.

[6] HAN L, LIU J, ZHANG D, *et al*. A portable TPM scheme for general-purpose trusted computing based on EFI[A]. Proceedings of the 5th International Conference on Multimedia Information Networking and Security[C]. Beijing, China, 2009.140-143.

[7] ZHANG D, HAN Z, YAN G. A portable TPM based on USB key[A]. Proceedings of the 17th ACM Conference on Computer and Communications Security[C]. Chicago, USA, 2010.750-752.

[8] EKBERG JE, BUGIEL S. Trust in a small package: minimized MRTM software implementation for mobile secure environments[A]. Proceedings of the 4th ACM Workshop on Scalable Trusted Computing[C]. Chicago, USA, 2009.9-18.

[9] DIETRICH K. An integrated architecture for trusted computing for java enabled embedded devices[A]. Proceedings of the 2nd ACM Workshop on Scalable Trusted Computing[C]. Alexandria, USA, 2007.2-6.

[10] DIETRICH K, WINTER J. Towards customizable, application specific mobile trusted modules[A]. Proceedings of the 5th ACM Workshop on Scalable Trusted Computing[C]. Chicago, USA, 2010.31-40.

[11] WINTER J. Trusted computing building blocks for embedded linux-based ARM trustzone platforms[A]. Proceedings of the 3rd ACM Workshop on Scalable Trusted Computing[C]. Alexandria, USA, 2008.21-30.

[12] SANTOS N, RAJ H, SAROIU S, *et al*. Using ARM trustzone to build a trusted language runtime for mobile applications[A]. Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems[C]. Salt Lake City, UT, USA, 2014.67-80.

[13] WU C, ZHOU Y, PATEL K, *et al*. AirBag: boosting smartphone resistance to malware infection[A]. Proceedings of the 21th Annual Network and Distributed System Security Symposium[C]. San Diego, California, USA, 2014.

[14] GILAD Y, HERZBERG A, TRACHTENBERG A. Securing smartphones: a micro-TCB approach[J]. IEEE Pervasive Computing Magazine, 2014.

[15] EISENBARTH T, GÜNEYSU T, PAAR C, *et al*. Reconfigurable trusted computing in hardware[A]. Proceedings of the 2nd ACM Workshop on Scalable Trusted Computing[C]. Alexandria, USA, 2007.15-20.

[16] SCHELLEKENS D, TUYLS P, PRENEEL B. Embedded trusted computing with authenticated non-volatile memory[A]. Proceedings of the 1st International Conference on Trusted Computing and Trust in Information Technologies[C]. Villach, Austria, 2008.60-74.

[17] ARENO M, PLUSQUELLIC J. Securing trusted execution environments with PUF generated secret key[A]. Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications[C]. Liverpool, UK, 2012.1188-1193.

[18] BERGER S, CACERES R, GOLDMAN KA, *et al*. vTPM: virtualizing the trusted platform module[A]. Proceedings of the 15th Conference on USENIX Security Symposium[C]. Vancouver, Canada, 2006.305-320.

[19] ENGLAND P, LOESER J. Para-virtualized TPM sharing[A]. Proceedings of the 1st International Conference on Trusted Computing and Trust in Information Technologies[C]. Villach, Austria, 2008.119-132.

[20] STUMPF F, ECKERT C. Enhancing trusted platform modules with hardware-based virtualization techniques[A]. Proceedings of the 2nd Second International Conference on Emerging Security Information, Systems and Technologies[C]. Cap Esterel, France, 2008.1-9.

[21] CHEN C, RAJ H, SAROIU S, *et al*. cTPM: a cloud TPM for cross-device trusted applications[A]. Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation. Seattle, WA, USA, 2014.187-201.

[22] Trusted Computing Group. TCG software stack (TSS) specification version 1.2[EB/OL]. http://www.trustedcomputinggroup.org, 2014.

[23] TPM Emulator. Software-based TPM emulator[EB/OL]. http://tpm-emulator.berlios.de, 2014.

[24] Lynn B. PBC library–the pairing-based cryptography library[EB/OL]. http://crypto.stanford.edu/pbc, 2014.

[25] IBM's software TPM. IBM software trusted platform module[EB/OL]. http://ibmswtpm.sourceforge.net, 2014.

[26] Digia. Qt product[EB/OL]. http://qt.digia.com/Product, 2014.

[27] Thesycon. USB CDC/ACM class driver for Windows 8, 7, Vista, XP[EB/OL]. http://www.thesycon.de/eng/usb_cdcacm.shtml, 2014.

作者简介：

张倩颖（1986-），女，河北三河人，中国科学院博士生，主要研究方向为网络与系统安全、可信计算。

赵世军（1985-），男，山东潍坊人，中国科学院博士生，主要研究方向为网络与系统安全、可信计算。

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

秦宇（1979-），男，重庆人，博士，中国科学院助理研究员，主要研究方向为网络与系统安全、可信计算。

冯登国（1965-），男，陕西靖边人，中国科学院研究员、博士生导师，主要研究方向为网络与信息安全。