软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(5):1392−1405 [doi: 10.13328/j.cnki.jos.005953]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于 TEE 的主动可信 TPM/TCM 设计与实现*

董 攀[1], 丁 滟[1], 江 哲[2], 黄辰林[1], 范冠男[1]

[1](国防科技大学 计算机学院,湖南 长沙 410073)

[2](Department of Computer Science, University of York, YO10 5GH, UK)

通讯作者: 丁滟, E-mail: dingyan_ding@aliyun.com

摘 要: 可信技术正在从被动可信度量向着下一代的主动可信监控方向发展,要求 TPM/TCM 模块有能力主动度量和干预主机系统,传统的 TPM/TCM 从架构和运行机制等方面都无法满足这种能力.TEE(trusted execution environment)技术提供了可信执行环境和主动访控能力,为构建下一代 TPM/TCM 提供了基本平台,但还存在系统结构、存储以及通信等多方面挑战.提出了基于 ARM 平台 TrustZone 机制的 TZTCM(TrustZone-based trusted cryptography module)方案,通过分核异步系统架构解决 TZTCM 独立可信运行和主动可信安全监控问题,基于 PUF(physical unclonable functions)安全存储机制和基于 UUID(universally unique identifier)的 TEE 安全通信机制,解决了 TEE 环境下可信平台模块的存储安全和通信安全问题,为设计实现主动可信 TPM/TCM 给出了理论和实践参考.通过实验验证了所提关键机制的有效性,实验结果表明,TZTCM 在密码计算能力上较常见 TPM 也有很大提升.TZTCM 只需要在系统中增加或修改相应的软/固件,除了主动可信监控能力,还具有低成本、高性能、低功耗、易升级等特点,相对传统 TPM/TCM 具有非常明显的优势.

关键词: TPM;TCM;主动可信;TrustZone;TEE

中图法分类号: TP311

中文引用格式: 董攀,丁滟,江哲,黄辰林,范冠男.基于 TEE 的主动可信 TPM/TCM 设计与实现.软件学报,2020,31(5):1392−1405. http://www.jos.org.cn/1000-9825/5953.htm

英文引用格式: Dong P, Ding Y, Jiang Z, Huang CL, Fan GN. Design and implementation of TPM/TCM with active trust based on TEE. Ruan Jian Xue Bao/Journal of Software, 2020,31(5):1392−1405 (in Chinese). http://www.jos.org.cn/1000-9825/5953.htm

Design and Implementation of TPM/TCM with Active Trust Based on TEE

DONG Pan[1], DING Yan[1], JIANG Zhe[2], HUANG Chen-Lin[1], FAN Guan-Nan[1]

[1](College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China)

[2](Department of Computer Science, University of York, YO10 5GH, UK)

Abstract: Trusted computing is being developed towards the next-generation active protection and monitoring, which requires that the TPM/TCM has the ability to actively measure and intervene the host system. Unfortunately, traditional TPM/TCM cannot satisfy the requirements in the respects of the architecture and the runtime mechanisms. Trusted execution environment (TEE) technology provides a trusted execution environment and the ability of accessing/controlling the host resources during the run-time, which brings a foundation for the next generation TPM/TCM. However, there are still three main challenges: software architecture, secure storage, and secure

* 基金项目: 国家重点研发计划(2018YFB0803501); 核高基国家科技重大专项(2017ZX01038104-002); 国家自然科学基金(61602492, 61303191, 61502510, 61872444)

Foundation item: National Key Research and Development Program of China (China) (2018YFB0803501); CHB National Science and Technology Major Project of China (2017ZX01038104-002); National Natural Science Foundation of China (61602492, 61303191, 61502510, 61872444)

本文由“系统软件构造与验证技术”专题特约编辑赵永望副教授、刘杨教授、王戟教授推荐.

收稿时间: 2019-08-30; 修改时间: 2019-10-24; 采用时间: 2019-12-24; jos 在线出版时间: 2020-04-07

communication. This study proposes the design and implementation of TZTCM (TrustZone-based trusted cryptography module), which is a TPM/TCM scheme based on ARM TrustZone. TZTCM adopts several key mechanisms to overcome the three challenges. Firstly, the non-uniform core assigned and asynchronous (NUCAA) system architecture is designed to enable the independent and active operation of TZTCM. Secondly, the secure storage mechanism based on physical unclonable functions (PUF) is designed to guarantee the privacy of data in TZTCM. Thirdly, the secure communication mechanism based on universally unique identifier (UUID) is designed to prevent the channel (between host and TZTCM) from malicious activities. Therefore, TZTCM provides a prototype system of the next-generation TPM/TCM. It is shown that TZTCM has the identical security as a hardware TPM/TCM chip via theoretical analysis. An instance of TZTCM is implemented on an ARM development board (Hikey-board 620), and the runtime test shows that TZTCM can achieve higher performance for cipher computing than traditional TPMs. Compared to current TPMs/TCMs, TZTCM has obvious advantages in many aspects: active safeguard capability, only software/ firmware required, easy update, and low power consumption.

Key words: TPM; TCM; active trust; TrustZone; TEE

在以传统 TPM/TCM(国外使用 TPM 标准,即 trusted platform module;国内采用 TCM 标准,即 trusted cryptography module)为根基建立的可信 2.0 架构[1]中,可信基础部件被设计为在硬件隔离的环境中以被动方式工作,供主机 CPU 进行服务调用.这种方式在安全上不完备,存在以下弊端:难以保证加载时(特别是加电时)的软件完整性;软件更新困难;无法保证在运行态的完整性.针对这些弊端,我国已在可信计算技术 3.0 阶段提出“主动防御体系”思想,目标是确保全程可测可控、不被干扰,即防御与运算并行的“主动免疫计算模式”[1].然而受架构所限,传统的 TPM/TCM 构建技术不具备对主机系统进行主动访问和监控的能力[2],甚至难以掌控系统上电时的代码可信性.此外,传统 TPM/TCM 还在应用中表现出多方面问题,由于采用廉价芯片导致性能普遍偏低,物理芯片在成本、功耗、散热等多方面不利因素较多,芯片封装不易升级维护.随着技术的快速发展,这些问题变得越来越突出,迫切需要在 TPM 的结构实现方面予以创新.TEE 扩展技术提供了这种可能,例如 ARM 处理器中的 TrustZone 技术以及 Intel 处理器中的 SGX(software guard extensions)技术,都能为计算平台提供一个隔离于平台其他软硬件资源的运行时环境.其中,TrustZone 技术被设计为在系统加电后优先获得控制权,并拥有比主机更高的访问和控制权限,因此更为贴合 TPM/TCM 的功能和安全性需求.

如图 1 所示,TPM/TCM 的各项服务机制可以映射到 TrustZone-TEE 的基本结构中.在 TEE 中封装虚拟 TPM/TCM 核心服务,包括密码算法引擎、命令执行引擎以及安全持久存储等.在主机中利用标准 TEE Client API 封装由 TDDL 层所调用的 TPM/TCM 命令,对 TSS 的 TDDL 以上层次保持透明.将 TPM/TCM 命令被封装为标准 TEE 通信协议,通过 SMC 服务接口调用核心 TPM/TCM 服务.例如,Raj 等人基于这种思想提出了基于 TrustZone 的固件化 TPM 实现方法[3],验证了通过 TrustZone 实现低成本 TPM 的可行性.

主动可信能力被定义[1]为:“可信节点通过底层的监控点,以主动监控的方式监视系统的行为,并通过信息系统整体的策略管控,构建可信计算体系,为应用创建一个安全保障环境,确保应用按照预期执行,免于黑客、病毒等威胁”.TrustZone 虽然提供高于主机系统的访问权限,但要实现主动可信能力还面临诸多难点,本文总结为 3 项挑战.

- 一是 TPM/TCM 与主机的“协同结构”问题.TEE 的设计思想是与主机分时共享 CPU,隔离使用存储和 IO.受主从式架构设计的制约,已有的 TrustZone 基础服务大多是以被动执行的思想设计,如要提供主动运行和资源访问能力,则需要设计较为复杂的主机-TEE 切换机制,并且要避免 TEE 过久“占领”CPU,防止主机系统的中断等机制受到超时等因素的影响.
- 二是 TrustZone 的安全持久存储机制能力不足,借用 REE(rich execution environment)存储或者单独提供安全态的存储接口,无法阻止恶意删除或者使用物理手段探测私密数据.
- 三是通信安全问题.传统 TPM 上已经存在多种链路攻击手段,例如 reset 攻击和中间人攻击[4],可能对 TPM/TCM 安全性产生致命影响.基于 TrustZone 中的虚拟 TPM/TCM 尽管不依赖物理总线通信,但是信道安全问题依然存在,恶意软件可以通过提权攻击等方式直接干涉主机与 TEE 环境的底层通信,对虚拟 TPM/TCM 的通信安全构成威胁.

此外,为了提供实时主动的可信服务,TPM/TCM 必须要有足够强大的计算能力,以实现高效的动态度量.

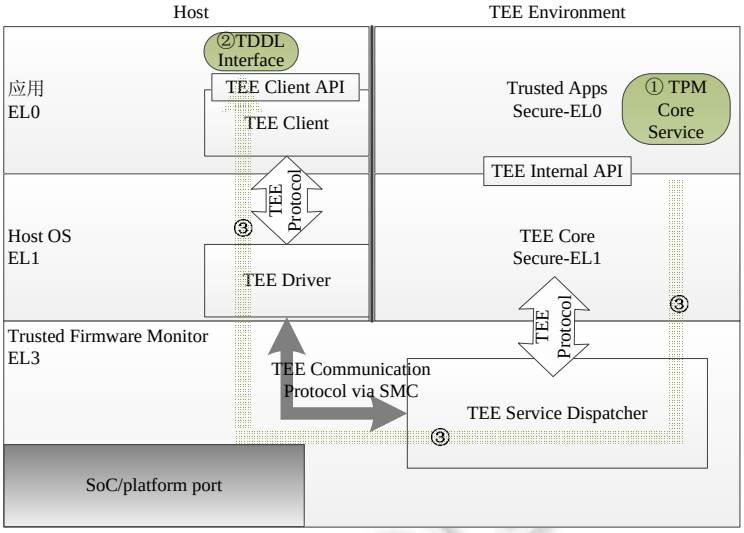

Fig.1 Principle of virtual TPM/TCM based on TEE

图 1 基于 TEE 的虚拟 TPM/TCM 基本设计思想

本文针对以上挑战,在系统架构、存储机制以及通信机制等方面实现了突破,提出了 TZTCM(TrustZone-based trusted cryptography module)构建方案,包括分核异步系统架构、基于 PUF(physical unclonable functions)安全存储和基于 UUID(universally unique identifier)的安全通信.除了对传统 TPM/TCM 功能的兼容以外,TZTCM 还具备主动可信能力,具体包括:(1) TZTCM 能够对度量根进行隔离和加电时的首先验证;(2) TZTCM 能够对通过静态度量后的运行态软硬件进行实时的再度量;(3) TZTCM 可以对运行中的主机系统进行干预.通过理论分析证明了 TZTCM 拥有与传统 TPM 相当的安全防护能力,通过实验证明 TZTCM 拥有远高于商业 TPM 器件的密码计算能力.由于是硬件机制支持下的软件实现,TZTCM 在成本和升级维护性方面具有传统 TPM/TCM 无法比拟的明显优势.

本文第 1 节介绍 TPM/TCM 改进工作的相关进展.第 2 节提出基于 TEE 的 TZTCM 设计与构建方案,解决 3 个关键的挑战问题.第 3 节论述 TZTCM 主动可信的实现机理.第 4 节对 TZTCM 的安全性进行全面的对比分析.第 5 节介绍对 TZTCM 的运行测试和性能验证.最后是总结和未来工作展望.

1 相关研究工作

1.1 可信基础模块TPM和TCM

可信基础模块始于 2000 年可信计算平台联盟(trusted computing platform alliance)制定的 TPM1.0 规范.2003 年,TCG(trusted computing group)成立,修改完成了 TPM1.1 规范,2004 年发布了 TPM1.2,2014 年发布了 TPM2.0 规范.鉴于可信计算技术对国家信息安全体系的重要性,经国家密码管理局批准,中国于 2006 年成立了可信计算密码专项组,并于 2008 年 12 月更名为中国可信计算工作组(China TCM Union),简称 TCMU.2007 年 12 月,国家密码管理局颁布了《可信计算密码支撑平台功能与接口规范》,将国内使用的可信基础模块定义为 TCM(trust cryptography module).相较于 TPM,TCM 采用了我国《商用密码管理条例》中规定的 SM2、SM3 等国密算法,同时引入了对称密钥算法,简化了 TPM 中复杂的密钥管理.TCM 的证书认证机制采用签名密钥以及加密密钥的双证书机制,将对称密钥与非对称密钥结合保护系统安全,在密钥管理体系和基础密码服务体系等方面进行了改进,提升了系统的安全性.

TPM 和 TCM 的构成和功能类似,提供可信计算平台的信任根(RTS,RTR),是由 CPU、存储器、I/O、密码协处理器、随机数产生器和嵌入式操作系统等部件组成的独立 SoC 芯片,具备可信度量的存储、可信度量的报

告、密钥产生、加密和签名、数据安全存储等功能.由于信息安全应用需求的不断变化,基于 TPM/TCM 的信任链方案已经不能满足应用需求.存在的问题有[5]:信任链传递方案存在安全隐患,如 BIOS 可擦写;针对度量过程的时间差攻击;TPM 与可信平台的通讯数据可被窃取;可信根缺乏对硬件设备的控制权;Reset 攻击重置平台配置寄存器(PCR);CPU 和内存均可能被攻击等.针对上述问题,我国于 2016 年提出了可信平台控制模块(trusted platform control module,简称 TPCM)[1]的概念,要求能够直接接入平台的主设备接口,具备对软硬件系统的可信控制、主动度量和主动报告等功能,以便主动监控平台各组件的完整性和工作状态.TPCM 的主动可信体系对 SoC、系统总线以及主板设计都提出很高的要求,受限于硬件设计和制造能力,目前还没有完全具备设计要求的成品器件[2].

1.2 TPM/TCM改进研究

在应用中发现,传统 TPM/TCM 器件不仅存在安全缺陷,而且存在成本高、难以升级等问题,研究者在多个方向上进行了改进探索.

为了增强已有 TPM/TCM 的主动度量能力,文献[2]提出通过主板改造,使 TPM/TCM 能够通过仲裁器和 CPU 互斥的访问主机内存和 flash,在运行中度量作为可信软件基(TSB)核心的基础可信基(TBB),TBB 度量 TSB 中的其他部件,从而实现了一种简化的 TPCM 方法.类似的思想还在文献[6]中发展为嵌入式系统中的 FPGA 实现.该方法实现简单,但只对内存有访控能力,并且在动态度量中存在遭遇恶意干扰的可能.

由于可信路径的建立有严格的时序限制(PCR 寄存器的更新特性)和单向性,TPM/TCM 无法直接满足为虚拟化架构中的多虚拟机同时提供可信根的需求,TPM2.0 方案中提供 DRTM[7]支持在机器启动的任何时刻创建可信根,但必须借助 CPU 的硬件支持,例如 Intel 的 TXT 或者 AMD 的 SVM.DRTM 的原理是由操作系统通过特殊指令创建动态可信链,重置动态 PCR 寄存器到缺省值并开始验证流程,动态度量的第 1 步是由 CPU 度量一个签名的硬件模块(TXT 或 SVM 技术).另一种为虚拟化提供可信支持的思想[8]是采用软件方式构建多个虚拟 TPM 实例(vTPM),由经过度量的 hypervisor 充当 vTPM 宿主,并向虚拟机提供树状可信路径支持.该思想已被 TCG 纳入“虚拟可信平台架构规范(virtualized trusted platform architecture specification)”.然而,由于 hypervisor 是一种纯软件实现,不能免除被恶意篡改的风险,削弱了 vTPM 作为可信根的安全性.

为了降低 TPM 硬件和可信软件栈的开发测试成本,以及提供 x86 平台外的可信技术验证,瑞士苏黎世联邦理工学院(ETH)和 IBM 公司分别开发了 TPM Emulator[9]和 Software TPM[10]用于 TPM 的软件模拟.由于缺少安全运行环境的支持,软件 TPM 一般仅用于示范教学或者方案论证,无法用于可信计算生产系统.

为了增强 TPM 的灵活性和可迁移性,文献[11]提出一个便携式 TPM 方案,在一个具备 USB 接口以及 EFI 接口的加密芯片上实现 TPM 的功能,具备与传统 TPM 同等水平的安全保护能力.该方案中的证书和密钥既可以与平台绑定,又可以针对不同的平台和用户提供灵活便携性.出于同样的目的,文献[12]提出一种基于 FPGA 实现 TPM 的方案.借助于 FPGA,TPM 拥有更新和加入新的密码模块的能力.可变硬件 TPM 方案能够支持 TPM 的升级和多平台复用,但以增加硬件成本为代价,而且器件本身不与机器绑定,反而会增加用户的安全顾虑.

综合分析可以看出,由于 SoC、总线以及主板设计的局限性,对传统硬件 TPM/TCM 进行功能增强的方式遇到了较大的障碍,软件化的 TPM/TCM 功能改进又引入了额外的安全风险,折中的方法只有通过软硬结合的方式设计新型的 TPM/TCM.

1.3 主动防御技术

主动防御是指在指令执行的同时进行安全防护,目标是全程可测可控[1],其实现方式主要分为基于软件与基于硬件两类.软件方式可通过基于内核进行强制的代码检测[13,14]或者引入一个更高特权级的软件层[15,16],例如 hypervisor.这类方法的共有缺陷是因为频繁的现场切换而引入较明显的性能损耗,由于是软件实现的,其本身也容易受到恶意侵害.保证新增软件模块的可信性本身也引入了新问题,这种方法也难以融入到可信 3.0 模式的可信部件设计中.基于硬件的方法[17−20]一般会与主机系统进行物理隔离,利用硬件所提供的一些事件触发机制探测异常事件或者主机状态的改变.硬件方法虽然能够对主机中的恶意代码起到很好的免疫作用,并且几

乎不引入额外开销,但却存在语义鸿沟和探测能力不足的问题.这是因为主动防御要求扩展硬件能够对主机软件运行期间的各类事件进行监控.第一,能够感知主机系统的内存访问事件,这是因为内核使用内存保留其状态信息和敏感数据结构;第二,应能够监视系统的状态寄存器,这对于理解系统的当前配置和资源配属非常重要;第三,外部机制应有能力控制和改变主机系统的关键寄存器和内存空间,以阻止恶意行为并恢复系统的正常.由于系统的复杂性和总线链接的限制,已有的硬件扩展机制还很难满足这 3 种需求.

1.4 TrustZone及相关解决方案

TrustZone[21,22]是 ARM CPU 的 TEE 安全扩展机制,能够保证安全态软件在加电时首先启动,并对后续加载的启动映像进行逐级验证.TrustZone 使能后,物理处理器能够在两种安全模态之间切换,分别定义为常态(normal world,运行主机 OS)和安全态(secure world,运行 TEE OS).TrustZone 提供了完善的隔离能力,对于资源的访问许可严格受控于安全态,安全态的资源禁止被常态软件所访问.模态切换功能由 SMC(secure monitor call)指令负责实现.TrustZone 技术推荐将安全资源封装在 SoC 芯片内部,以防止利用引脚进行物理窥探.但由于技术和成本限制,SoC 内一般不会封装大容量的永久存储.很多解决方案推荐使用 eMMC 存储器的 RPMB (replay protected memory block)分区实现大量数据的安全读写和存储.为了方便常态和安全态的软件开发,ARM 公司和 GlobalPlatform 组织都制定了相应的软件规范.此外,还有相关研究[23,24]通过在 IO 硬件中加入虚拟隔离技术的方法,也能为安全态提供隔离存储能力.但这种方法没有考虑直接对物理存储进行攻击和读取的防护,故不适用于虚拟 TPM/TCM 的存储支持.

为了保证可信执行环境的完备性,TrustZone 技术包含了 TBBR(trusted board boot requirements)子规范[25],特别定义了系统从加电之后的可信保障流程以及软硬件需求.TBBR 规定系统必须有一个信任根作为一个信任起点(例如 SoC one-time-programmable(OTP)存储器中的公钥),从 reset 加电之后立即发挥作用,通过可信根建立可信链,并逐步验证:进一步的签名方式、启动代码、TEE 环境、TEE 服务以及常规主机环境等.

Raj 等人提出了一种基于 TrustZone 实现固件化 TPM 的 fTPM 方案[3].fTPM 基于 3 种方法,满足 TrustZone 在 TPM 功能上的支持:增加额外的硬件,采用 eMMC 存储器作为 TPM 的持久安全存储设施;做出一些不损害 TPM 安全性的折中,例如限制密码计算的规模;修改 TPM 规范的部分语义,以适应 TrustZone 的一些局限.然而,fTPM 的设计没有考虑主动的可信度量和控制问题,并且忽视了通信和存储过程中的一些较为严重的安全挑战.本文将通过 TZTCM 的设计来解决这些问题.

2 基于 TEE 的 TZTCM 设计与构建

TZTCM 仍然采用图 1 所示的基本设计思想,即将 TPM/TCM 的基本功能和服务封装在 TrustZone-TEE 环境中,围绕简介部分总结的 3 个挑战问题,本节将提出对应的核心解决方案,通过设计分核异步 TZTCM 系统架构实现可信模块的独立运行和对主机系统的主动实时度量能力,通过基于 PUF 的 TEE 多层安全存储解决可信模块的持久存储安全性以及第一时间的可信加载问题,通过基于 UUID 的 TZTCM 安全通信解决可信模块的信道安全问题,最终保证 TZTCM 作为主动可信基础模块的功能实现和安全运行.

2.1 分核异步TZTCM系统架构

TrustZone 的 TBBR 规范使 TZTCM 以及系统加电时的完整性得到保证,为了具备传统 TPM/TCM 功能的兼容能力和其他主动可信能力,在结构方面必须考虑 3 个方面的问题:一是在线实时度量能力要求 TZTCM 拥有自主的任务分时调度;二是高负荷服务计算(如加解密)不应影响主机系统的正常执行;三是 TZTCM 拥有权限对主机系统的内存资源和 IO 资源进行控制.

针对第 1 个问题,已有的 TZDKS 等方案[26]提供了 TEE 环境的实时调度能力,但由于主机操作系统在调度上独立于 TEE 环境,因而多采用 Idle-scheduling 策略,即当 TEE 系统内有就绪任务时,主机操作系统将一直被挂起,直到所有 TEE 内的任务执行完毕.这样就使主机操作系统的时钟可控度变差,甚至严重影响后者的运行.由于 TEE 难以快速获知主机系统调度信息,通过调度策略解决该问题会增加 TEE 系统复杂度,对其服务能力造成

负面影响.针对第 2 个问题,可考虑利用 CPU 普遍拥有的多核特性,以专用的核进行 TZTCM 以及其他大计算量的服务,该核不被主机操作系统调度,因此不会对后者有任何影响.这种模式会要求新的 TEE 服务调用模式,即从同步的 CPU 核内调用改为异步的核间调用.针对第 3 个问题,,需要由 EL3 态的软件将主机侧的资源访问权限开放给 TEE 内的 TZTCM 代码,由后者实现运行时动态度量和控制的职能.其中,主机系统中的进程或者运行模块具体特征(如内存地址分布等)不易直接获取,可由经过度量的主机系统中的 agent 软件配合获得.

本文提出 TZTCM 主动可信功能所依托的分核异步 TEE-TCM 系统架构,如图 2 所示.

- 首先是 TZTCM 的分核执行特点.在多核 CPU 平台上,每个 core 都拥有受 Monitor 控制的 N-S 状态切换能力,其中,在 N 态运行常规的主机操作系统 REE 环境,在 S 态运行 Monitor 和 TEE 环境.TZTCM 的服务主体运行在 CPU 的一个独立 core 上(记为 s-core),s-core 被配置为不会切换到 N 态,因此主机操作系统并不感知 s-core 的存在.S-core 上拥有独立的 TEE 时钟 timer,驱动调度程序分时运行 TZTCM 的主动度量和监控模块.事实上,TZTCM 与主机操作系统一起以分核方式运行.
- 其次是 TZTCM 的异步服务特点.主机系统的可信调用请求先通过 SMC 调用经由 Monitor 模块转发到本核上运行的 TEE 服务代理,后者在触发一个目标是 TZTCM 所在 s-core 的软中断(SGI)后,会立刻执行服务调用返回,所在 core 会切换到主机操作系统环境,调用方在感知调用返回后立即休眠等待.TZTCM 收到软中断通知后进行服务的计算,完成后用软中断方式通知主机系统中的服务调用者,接收服务结果.因此,与传统的 TPM 服务类似,TZTCM 的服务以异步方式实施.在其他 core 上,TEE 中只有服务接口,并以被动方式提供安全服务.这些服务仍采用同步方式,由主机操作系统在任一 core 上发起 SMC 调用,经由 Monitor 代码转发到 TEE 服务,服务完成后将结果返回给调用者,所有指令都在同一个 core 上执行.

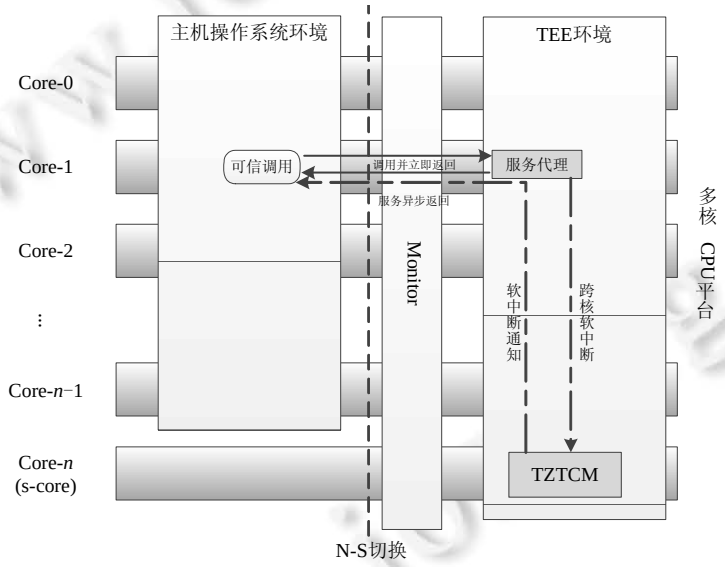

Fig.2 NUCAA (non-uniform core assigned and asynchronous) system architecture

图 2 分核异步系统架构

借助分核运行方式,TZTCM 的调度机制得到简化,更易于部署对主机系统模块的在线动态度量和监控,并免除了大计算量的服务对主机系统的影响;借助异步服务请求机制,服务调用保持与传统 TPM/TCM 架构兼容,解决了同步服务无法避免的主机系统挂起问题.

2.2 基于PUF的TEE多层安全存储

需要解决两个方面的安全问题:一是防止来自 TEE 外部的非法访问,包括运行时访问与静态物理窥探两类;

二是防止来自 TEE 中其他软件模块的在线非法访问.TZTCM 通过两种机制解决这些问题:1) 基于 PUF 的 RPMB 密钥保护,防止运行时的密钥外泄,并支持密钥更新;2) 层级加密存储方案,防止 TEE 中来自其他应用的非法窥探.

2.2.1 基于 PUF 的 RPMB 密钥保护

PUF(物理不可克隆函数)是硬件形式的单向函数,是物理定义的“数字指纹”,作为微处理器等半导体设备的唯一标识.它们基于半导体制造过程中自然发生的独特物理变化生成,能够用于实现安全功能,如设备认证、密码协议的密钥生成、为随机数生成器生成种子等等.目前已有很多 PUF 可以实用,例如基于器件延迟的 PUF、基于访存随机性的 PUF 等.利用 CPU 片内 SRAM 上电初始值的 PUF 特性[27],不但能够获取上电时的初始密钥,而且可以在读取密钥后通过对 SRAM 的写操作防止后续对该密钥的读取,有“阅后即焚”的效果.

TZTCM 对于 RPMB 的密钥使用有下述需求:密钥与安全 OS 映像分离,使用不安全的外部存储,可在 eMMC 卡更换时对应更换,从外部存储的信息无法获得密钥信息.因此,对 RPMB 密钥的保护不但要能抵抗来自 REE 环境的恶意代码攻击,还应抵抗通过实验室设备对 SoC 芯片的外部端口进行的探测和攻击.

TZTCM 对于 RPMB 密钥保护的基本思想是:在 SoC 外部持久存储设备中仅保存用于合成 Key_{RPMB} 的辅助数据 $Data_{KA}$,通过 PUF 技术进行 Key_{RPMB} 的生成和恢复;仅在 SoC 片内安全内存中以内核服务的方式使用 Key_{RPMB},对 TEE 应用仅保留加密或签名的接口.需要说明的是,并不需要对 $Data_{KA}$ 进行特别的安全保护.一方面暴露 $Data_{KA}$ 并不会造成 Key_{RPMB} 的暴露;另一方面,当有攻击者删除或是篡改 $Data_{KA}$ 后,能够被系统及时检测,从而终止后续操作.Key_{RPMB} 只在加电启动时有用,运行时并不存在因篡改导致的拒绝服务攻击风险.RPMB 密钥保护实现方案参考文献[27]中的 TrustZone 可信根设计,分为 Key_{RPMB} 的生成、恢复和更新 3 个部分,简记为 GRU(generate-rebuild-update).

- Key_{RPMB} 的生成过程是出厂时进行的,如图 3 上半部分所示.首先,随机选择一个种密钥 Key_S(seed key),利用 TPM 规范中所定义的 KDF(key derivation function)生成 Key_{RPMB},并写入 eMMC 存储器 RPMB 分区的 OTP Key 寄存器.之后对 Key_S 进行 BCH 编码变换,得到的结果与主 CPU 的 PUF 读取值进行异或操作(XOR),得到 $Data_{KA}$,最后销毁 Key_S,并将 $Data_{KA}$ 保存到设备的持久存储中.

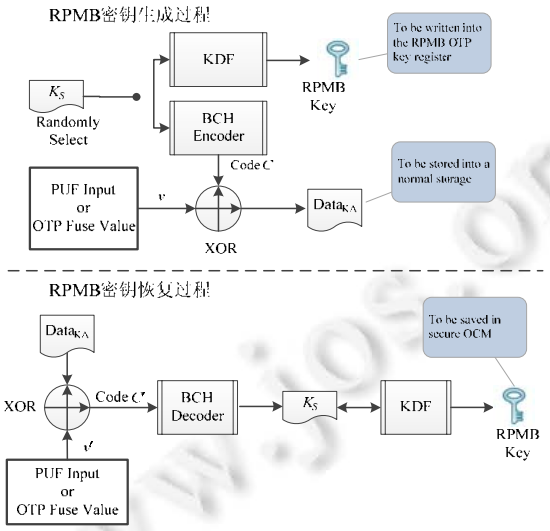

Fig.3 Generation and recovery processes of RPMB secret key based on PUF

图 3 基于 PUF 的 RPMB 密钥的生成和恢复

- Key_{RPMB} 的恢复用于每次设备重启,需要读取 RPMB 分区之前.如图 3 下半部分所示,使用主 CPU 的 PUF 读取值与常规存储的 $Data_{KA}$ 进行异或,得到的值进行 BCH 解码操作,即可得到 Key_S,再用 KDF 得

到 Key_{RPMB}.仅 Key_{RPMB} 和 TEE 的内核签名/验证函数共同保存在安全的 OCM(on-chip memory)存储器中,由后者对其他 TEE 应用提供计算结果.

- Key_{RPMB} 的更新用于设备因维修或其他原因需要更换 eMMC 存储器,其基本过程和生成过程一致.

利用 PUF 的特性,Key_{RPMB} 的有用信息暴露时间最小化.这是因为多数 PUF 值(如基于 RAM 的 PUF)的读取时机仅存在于系统加电之初.而通过芯片外部接口的静态读取、以及加电后的 REE 恶意代码、或者 TEE 应用中的恶意代码都无法读取到 Key_{RPMB} 的有用信息,因而增强了对 Key_{RPMB} 的保护.

2.2.2 采用层级加密存储的私密设计

为了保证 TZTCM 的存储数据不被非法窥探,在 TEE 层、应用层、文件层分别使用不同的密钥进行加密,并且这些密钥形成如图 4 所示的树形关系.这样可以对 TEE 中不同的应用及文件进行隔离,防止一个应用的数据被其他应用访问.

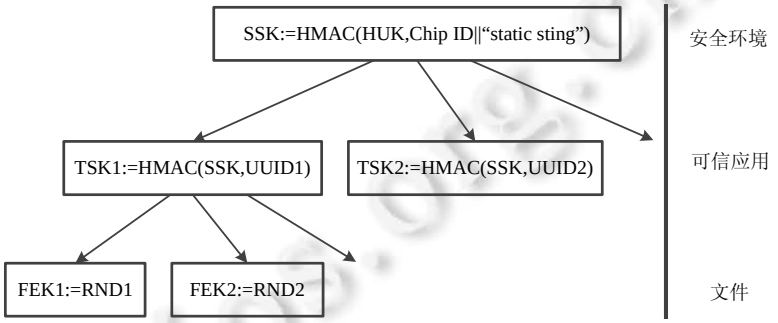

Fig.4 Secret key hierarchy for secure storage

图 4 安全存储的层级密钥体系

- TEE 层:安全存储密钥(SSK)用于生成每个可信应用层的存储密钥.当 TEE 启动时,使用相关函数读取硬件唯一码 HUK 以及芯片的 ID 号,并使用 HMAC 散列算法生成 128 位的 SSK.在 TEE 的生命周期中,SSK 一直存在于安全内存中,不会外泄到非安全区.
- 应用层:可信应用存储密钥(TSK)由 SSK 和通用唯一识别码(UUID)共同生成.每一个可信应用由 SSK 通过 HMAC 算法对 UUID 生成 128 位的摘要值作为 TSK.TSK 的主要作用是对属于自己的文件存储密钥(FEK)加密和解密.由于每个可信应用的 TSK 是不同的,这样可以保证可信应用无法解密属于其他可信应用的 FEK,进而保证了可信应用资源的独立性.
- 文件层:对于每一个文件,会生成一个 128 位的随机数作为文件存储密钥(FEK),用于对文件数据进行加密.该密钥经过所属可信应用的 TSK 加密后存入对应的文件配置表条目中,需要时使用相应 TSK 对其解密,这样能够保证其他可信应用无法读取该文件的内容,在一定程度上保证了数据的独立性.

基于这样的设计,TEE 中的文件存储可以抵御来自多个方面的威胁或攻击:(1) 即使攻击者拿到了 RPMB 分区的密钥,也无法解密 RPMB 文件系统;(2) 即使 TEE 系统中有不安全的应用,该应用可以读写 RPMB 分区,但也无法获得其他应用以及 TZTCM 的相关文件信息.

2.3 基于UUID的TZTCM安全通信

TPM 规范规定了 TPM 命令和数据的传输格式,使用 HMAC 保证数据的完整性和可信性.TZTCM 利用标准 TrustZone API 实现 TDDL 与服务端的通信,并基于两种机制实现通信链路的安全性[28]:(1) 会话;(2) 基于 UUID 的消息摘要.

UUID 是通用唯一识别码(universally unique identifier)的缩写,是一种软件建构的标准,亦为开放软件基金会组织在分布式计算环境领域的一部分.其目的是让系统中的所有模块都能有唯一的辨识信息,而不需要通过中央控制端来做辨识信息的指定.这样,每个模块都可以创建不与其他模块冲突的 UUID.

首先通过会话维护连接的完整性.图 5 展示了 TDDL 与 TZTCM 建立连接的过程:首先,TDDL 向 TEE Client 发送创建 Context 的请求,Context 被用来建立 TEE Client 和 TEE Core(内核)的逻辑连接,必须在创建会话之前初始化;然后,TDDL 向 TEE Client 发送建立会话的命令,通过底层中断和状态转换,该命令被传递给 TEE 内核,TEE 内核加载 TZTCM 业务服务,并执行创建会话的入口程序;最后,会话的状态会返回给 TDDL.在会话的基础上由基于 UUID 的散列值来保证安全性.TZTCM 作为一个 TEE 安全应用,拥有唯一的 UUID 标识.在初始化 TDDL 的可信关系时,应用安全密钥交换协议生成一个和 TDDL 共享的密钥 $K_{T\text{-}UUID}$,由 TDDL 和 TZTCM 分别秘密保存.在建立会话传递消息时,消息的生成方都会使用 $K_{T\text{-}UUID}$ 生成一个对消息散列值的签名,并附在消息结尾,用以鉴别消息的合法性.只有拥有 $K_{T\text{-}UUID}$ 的应用(如 TDDL)才可以与 TZTCM 建立会话,这保证了会话的可信性.当 TDDL 不再需要 TZTCM 的服务时,可以通过发送关闭会话命令关闭与 TZTCM 的连接会话.

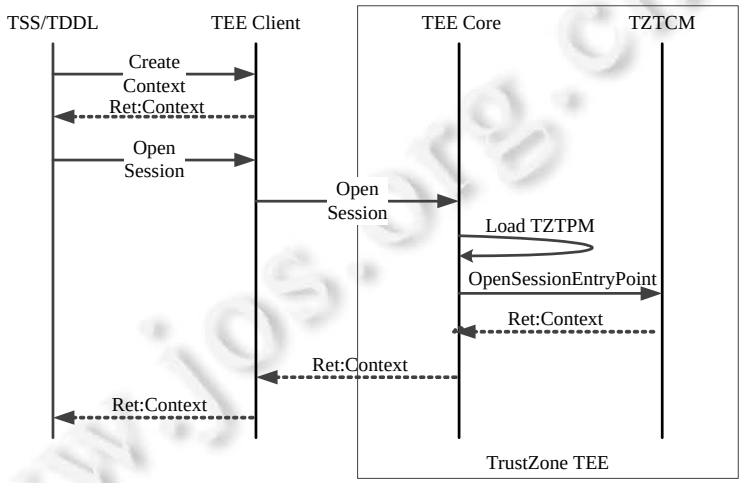

Fig.5 Process to establish a session between TDDL and TZTCM

图 5 TDDL 与 TZTCM 的会话建立过程

3 TZTCM 主动可信能力的实现

本节讲述 TZTCM 主动可信的 3 个方面能力的实现:加电时的主动可信根度量、运行时对主机系统的实时再度量、对运行中的主机系统主动干预.

3.1 系统加电完整性保证

基于第 2.2 节中的安全存储机制,可以为 TZTCM 建立以硬件可信根为基础的初始可信链.首先需要指出,TZTCM 的功能代码位于 TEE 的安全 OS 映像中.当设备加电时,CPU 默认处于安全态,并执行 BootROM 中的不可修改代码,后者是在芯片生产时一次性写入的,默认可信.BootROM 代码首先验证 RPMB 密钥的 GRU 代码的完整性(通过制造者公钥验证签名).然后,BootROM 读取 PUF 的初始值 v,初始化 OCM 并载入 GRU 代码.如果完整性检验成功,BootROM 将 v 传给 GRU 并在 OCM 中执行 GRU;否则,终止启动.

BootROM 代码通过 OTP fuse 中的公钥检查安全 OS 以及安全服务的完整性.如果验证成功,则加载安全 OS.在运行安全 OS 之前,BootROM 代码将初始化 PUF 对象中可变信息,相当于销毁操作.此时,TZTCM 的功能代码以及持久存储文件都得到了可信验证,初始可信链建立完毕.

3.2 运行时的实时度量

TZTCM 中的实时任务调度功能给了 TZTCM 主动发起度量的能力,发起对运行时的主机资源的实时度量.如图 6 所示,当主机系统启动时,TZTCM 通过传统的静态度量机制建立可信链,并在每一步更新 PCR 寄存器的

同时得到各目标的度量值记录在另外的数据库中.主机中将设立一个 Agent 任务,通过系统的一些信息库(如/proc 文件系统或者/boot 目录下的 System.map 文件)得到目标模块的地址信息传递给 TZTCM,TZTCM 利用 TrustZone 的访存权限对主机内存中的关键信息进行实时度量和监视,并可以根据静态度量的结果判断目标的当前可信性.本方案中,Agent 仅负责提供监控目标的物理地址,并不对目标进行实际监控.TZTCM 首先会在主机系统加载时验证 Agent 的静态可信性,并结合实时度量以及第 3.3 节中的动态措施保证 Agent 的运行可信性.

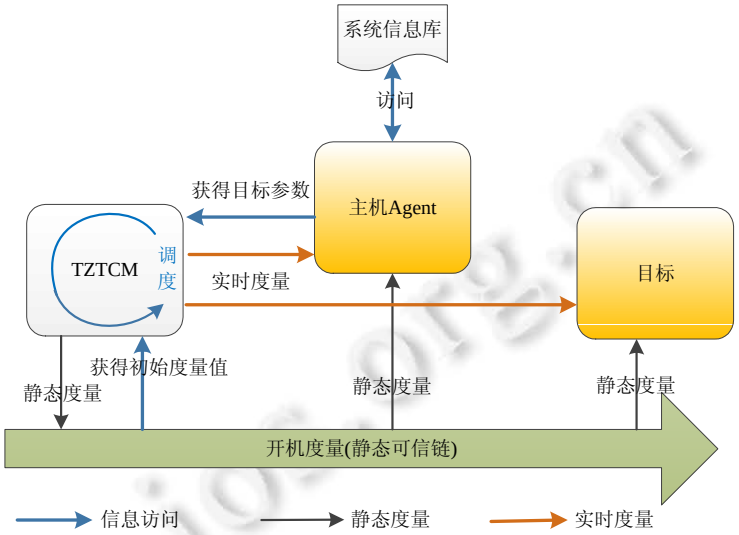

Fig.6 Real-time measurement mechanism of TZTCM

图 6 TZTCM 的实时度量机制

在 TZTCM 的实现中,我们特别将主机内核的代码段作为动态度量的目标,由 Agent 通过/proc/iomem 获得 linux 内核代码段数据段对应的物理地址,通过 TEE 环境提供的库函数实现了 TEE 内的虚地址映射,从而实现对代码段的动态读取和度量.

3.3 运行时的主动干预

TZTCM 不但能在主动度量发现可信异常时进行记录和报警,还可以对运行时的主机系统进行干预.

(1) 通过对 CPU 切换的控制,可以暂停主机系统的运行,或者利用通知方式使主机系统转入关机.

(2) 通过对内存的控制改写主机系统的内存映射页表,将可疑目标的内存设为不可执行(在页表描述项中,包含页面属性的定义,其中的 NX 位将定义页面中的代码是否被允许执行,TrustZone 拥有可访问 normal world 内存的权限,可以通过修改后者的页表属性设置特定内存不可执行),由主机系统做后续处理.尽管主机系统也有权限修改页面属性的 NX 位从而能够恢复页面的可执行,但 TZTCM 的监控将提升攻击的难度,并且能够通过监视已锁定的页面属性决定是否采取进一步措施.

(3) 通过修改控制器的权限剥夺对一些关键 IO 的读写操作,例如更改常态软件对 IO 地址的读写权限、或借助 TZPC(TrustZone protection controller)控制器直接控制特定 IO 控制器的归属等.

为了避免数据丢失和错误的服务,还需要通过进一步的逻辑设计利用这些主动干预能力.

4 TZTCM 安全性分析

TZTCM 和传统 TPM/TCM 可信模块拥有不同的结构,因而在安全性上也会有一定的差别.两者的结构差别表现如下.

- 通信:传统可信模块采用 LPC 总线或者其他低速总线与主 CPU 通信.TZTCM 不使用物理通信链路,仅

使用 TEE 软件通信方式.

- 内存:传统可信模块不使用主机内存,其内部存储对外隔离.TZTCM 使用 OCM 或者安全的片外内存.
- 存储:传统可信模块使用内部 NVRAM,对外隔离.TZTCM 使用 eMMC 存储器的 RPMB 分区.
- CPU:传统可信模块运行与主 CPU 无关.TZTCM 使用主 CPU,所涉及的寄存器和缓存与其他软件共存在一个 SoC 芯片上.

由上述对比可知,对 TZTCM 的攻击除了针对传统 TPM 的攻击类型外,可能还包括基于内存的攻击、基于存储的攻击以及基于 CPU 的攻击.

- 传统攻击.对传统 TPM 的攻击类型包括软件攻击、物理链路窃听和重置攻击、时间攻击等 3 类.其中,发自“主机系统”的软件攻击在传统 TPM/TCM 和 TZTCM 两种场景下是基本等效的.由于 TZTCM 不再暴露任何物理通信链路接口,因而物理链路窃听和重置攻击对于 TZTCM 无效.时间攻击主要针对 TPM1.2 版本之前的 RSA 算法,与本文 TZTCM 设计无关.
- 中间人攻击(含重放攻击).由于 TZTCM 会话需要验证消息签名,恶意软件虽然可以通过内核驱动接口访问 TZTCM,但会因为没有有效消息签名而遭到合法用户和 TZTCM 双方的拒绝,从而免除中间人攻击威胁.另外,在传统 TCM 通信协议中已经引入的“新鲜值”等机制同样适用于 TZTCM 中相关威胁的预防.
- 内存攻击.TZTCM 所使用的内存都由 TrustZone 进行保护,包括 DMA 访存模式,因而来自非安全态的内存窥探无法成功.然而,使用实验室仪器对设备接口处的管脚进行读取和扰动是无法防范的,在这方面,TZTCM 的安全性弱于传统独立 TPM/TCM 芯片.这种攻击的成本非常高昂,因此对于多数电子产品安全威胁来说不予考虑.
- 存储攻击.RPMB 分区的密钥仅在 OCM 中出现,通过软件攻击或是实验室仪器进行硬件攻击都无法获取该密钥,而且分区内的数据是分级加密存储的.因此,TZTCM 的存储安全性在某种程度上优于传统独立 TPM,这是因为利用开壳物理探测方式可以窥探到独立 TPM 中的明文数据.
- CPU 攻击.基于 CPU 中的寄存器或者缓存攻击窥探 TZTCM 的成本等价于针对 TrustZone 安全机制的攻击成本,因而本文不再单独讨论这种攻击的可行性.此外,TZTCM 中需要为 s-core 提供一个独立 timer,该 timer 可以利用 ARMv8 体系结构中为每个 core 所集成的 secure physical timer 实现.由于 secure physical timer 是 core 内的 IP,不存在管脚,因而拥有很强的安全性,不易受到非安全态软件或电气干扰.

近年来,有少量针对 TrustZone 的攻击能够威胁 TEE 系统的安全性,但这些攻击利用了 TEE 安全系统内核的漏洞.而我们知道,TEE 安全系统的复杂性和代码量远低于通用操作系统.因此,随着 TEE 软件系统的成熟和完善,这种攻击成功的可能性将会越来越低.同时,TEE 系统的灵活升级能力也会降低这种威胁.

5 TZTCM 实验测试

TZTCM 的实现和测试平台[28]以 Hikey 620 开发板为基础构建,使用 ARMv8 架构的麒麟 620 处理器,支持 ARM TrustZone 安全硬件技术,拥有主频 1GHz 的 8 个 core,2G 内存和 8G eMMC,支持 WIFI、蓝牙等功能.实验中使用 Linux 系统作为主机端的操作系统,内核采用经过修改的 Linux 4.4.0.本文选取 3 款市售典型商用 TPM 芯片进行性能对比,型号分别是 SSX35、ST19NP18、ST19WP18.由于 RSA 算法操作是 TPM 中较常用的操作,并且耗时最多,本文选择 2 048 位密钥 RSA 算法相关的操作来做性能比较实验.

图 7、图 8 显示了 TZTCM 和 3 款商用 TPM 芯片在签名和验证操作时间上的差异.可以发现,TZTCM 的性能比商用 TPM 芯片有很大提升.造成这种情况的主要原因是,TZTCM 使用了嵌入式设备的 CPU(ARMv8 架构 64 位处理器),而 TPM 芯片为了控制成本,使用的 CPU 性能并不强.因此在这些测试中,TZTCM 的性能有了巨大的提升.

为了验证 TZTCM 提供传统 TCM/TPM 服务时对 host 端系统的影响(即被动可信服务),系统 CPU core 被配置为都可在 host 与 TZTCM 之间切换的模式,使用 lmbench 工具对 TZTCM 系统进行测试,分别测试了 TZTCM

未运行状态下和 TZTCM 持续运行状态下 host 端系统的性能.为了方便对比,本文略去了基本无差别的测试项,仅保留了有明显差异的 fork proc、exec proc、sh proc、MMap Latency 等测试项,如图 9 所示.从测试结果可知,TZTCM 的被动运行对 host 端系统的运行性能影响几乎可以忽略.

在测试 TZTCM 主动防御能力的实验中,系统被配置为异步分核运行模式,即主动监控功能仅运行在 7 号 CPU core(s-core)上,主机系统只使用 0 号~6 号 CPU core.针对主机系统中执行的 Linux 内核进行主动监控,测试其代码段的可信性.在针对 Linux kernel 4.4 尺寸为 13.4MB 的代码段进行动态可信度量时,其单次度量时间约为 1.5 千万~2 千万 cycles,即发现可信异常的时间不超过 0.2s.当然,实际系统中的度量延迟还与度量任务的多少以及 TZTCM 的任务调度策略相关.通过 UnixBench 对引入主动可信服务前后的系统进行性能测试,结果显示,加入主动防御前后的总分为 384.6/374.7,性能下降约 2.57%.可见,TZTCM 主动防御对主机端系统的运行性能影响非常有限(如图 10 所示).

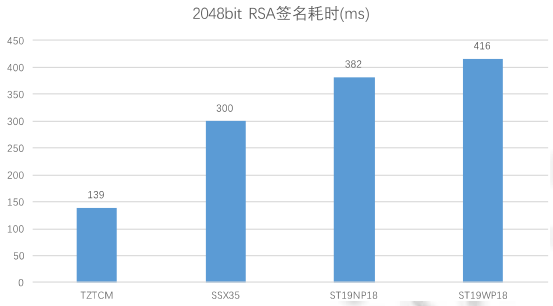

Fig.7 Comparison of signature operation time-consuming

图 7 签名操作时间比较

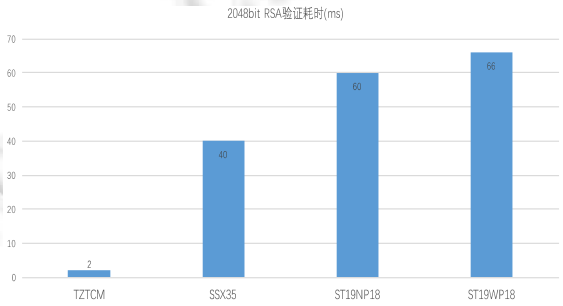

Fig.8 Comparison of verification operation time-consuming

图 8 认证操作时间比较

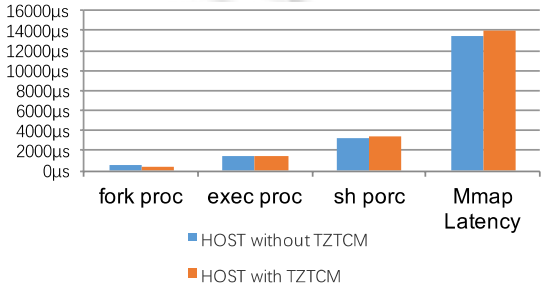

Fig.9 Performance influence of traditional trust service from TZTCM to the host OS

图 9 TZTCM 的传统可信服务对主机性能的影响

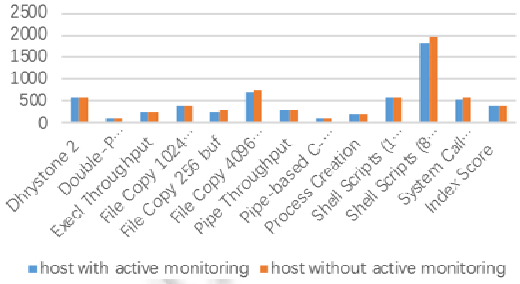

Fig.10 Performance influence of active monitoring service from TZTCM to the host OS

图 10 TZTCM 的主动可信监控服务对主机性能的影响

6 结 论

针对传统硬件 TPM/TCM 难以支持主动可信计算体系的问题,本文提出了基于 TEE 技术实现虚拟 TPM/TCM 的 TZTCM 方案.针对 TEE 虚拟 TPM/TCM 设计中面临的 3 项挑战问题,分别提出分核异步系统架构、基于 PUF 安全存储机制和基于 UUID 的 TEE 安全通信机制予以解决.TZTCM 具备 3 个方面的主动可信能力:加电时的主动可信根度量、运行时对主机系统的实时再度量、对运行中的主机系统主动干预.理论分析和实验结果表明,TZTCM 能够替代传统硬件 TPM 作为可信存储根和可信报告根为系统提供安全的可信服务,并拥有优于商业 TPM 器件的密码计算能力.在成本和升级维护性方面,TZTCM 更是具有传统 TPM 无法比拟的明显优势.

本文主要考虑了基于 TrustZone 实现主动防御型 TCM 的架构和关键技术问题,在 PUF 密钥管理、层次式安全存储协议、安全通信协议以及主动监控具体方法和实施策略方面还有大量问题有待解决,这些研究将在今后的工作中予以深入和推进.

References:

[1] Hu J, Shen CX, Gong B. Trusted Computing 3.0 Engineering Fundamentals. 2017 (in Chinese).

[2] Tian JS, Zhan J. Research and implementation of active dynamic measurement based on TPCM. Netinfo Security, 2016,(6):22−27 (in Chinese with English abstract).

[3] Raj H, Saroiu S, Wolman A, *et al*. fTPM: A firmware-based tpm 2.0 implementation. Technical Report, MSR-TR-2015-84, Microsoft Research, 2015. https://www.microsoft.com/en-us/research/publication/ftpm-a-firmware-based-tpm-2-0-implementation/

[4] Kursawe K, Schellekens D, Preneel B. Analyzing trusted platform communication. In: Proc. of the ECRYPT Workshop, CRASH-CRyptographic Advances in Secure Hardware. 2005.

[5] Guo Y, Mao JJ, Zhang CB, *et al*. Active measures based on a trusted platform control module. Journal of Tsinghua University (Sci & Tech), 2012,52(10):1465−1473 (in Chinese with English abstract).

[6] Wang X, Xu G, Han Y, *et al*. A trusted computing architecture of embedded system based on improved TPM. In: Proc. of the MATEC Web of Conf. on EDP Sciences, Vol.139. 2017.

[7] Nie C. Dynamic root of trust in trusted computing. In: Proc. of the TKK T1105290 Seminar on Network Security. 2007.

[8] Perez R, Sailer R, van Doorn L. vTPM: Virtualizing the trusted platform module. In: Proc. of the 15th Conf. on USENIX Security Symp. 2006. 305−320.

[9] Strasser M, Stamer H. A software-based trusted platform module emulator. In: Proc. of the Int'l Conf. on Trusted Computing. Berlin, Heidelberg: Springer-Verlag, 2008. 33−47.

[10] IBM. Software TPM introduction. http://ibmswtpm.sourceforge.net/

[11] Han L, Liu J, Zhang D, *et al*. A portable TPM scheme for general-purpose trusted computing based on EFI. In: Proc. of the 2009 Int'l Conf. on Multimedia Information Networking and Security. IEEE, 2009. 140−143.

[12] James MD. A reconfigurable trusted platform module. In: Proc. of the All Theses and Dissertations. 2017. https://scholarsarchive.byu.edu/etd/6298

[13] Song C, Lee B, Lu K, Harris W, Kim T, Lee W. Enforcing kernel security invariants with data flow integrity. In: Proc. of the NDSS. 2016.

[14] Chen Q, Azab AM, Ganesh G, Ning P. Privwatcher: Non-bypassable monitoring and protection of process credentials from memory corruption attacks. In: Proc. of the 2017 ACM on Asia Conf. on Computer and Communications Security. ACM, 2017. 167−178.

[15] Wang X, Qi Y, Wang Z, Chen Y, Zhou Y. Design and implementation of SecPod, a framework for virtualization-based security systems. IEEE Trans. on Dependable and Secure Computing, 2019,16(1):44−57.

[16] Vasudevan A, Chaki S, Jia L, McCune J, Newsome J, Datta A. Design, implementation and verification of an extensible and modular hypervisor framework. In: Proc. of the 2013 IEEE Symp. on Security and Privacy (SP). IEEE, 2013. 430−444.

[17] Lee H, Moon H, Heo I, Jang D, Jang J, Kim K, Paek Y, Kang B. KI-Mon Arm: A hardware-assisted event-triggered monitoring platform for mutable kernel object. In: Presented as Part of the 22nd USENIX Security Symp. 511−526. Washington: USENIX, 2013.

[18] Moon H, Lee J, Hwang D, Jung S, Seo J, Paek Y. Architectural supports to protect OS kernels from code-injection attacks and Their Applications. ACM Trans. on Design Automation of Electronic Systems, 2017,23(1):1−25.

[19] Koromilas L, Vasiliadis G, Athanasopoulos E, Ioannidis S. Grim: Leveraging GPUs for kernel integrity monitoring. In: Proc. of the Int'l Symp. on Research in Attacks, Intrusions, and Defenses. Springer-Verlag, 2016. 3−23.

[20] Kwon D, Oh K, Park J, Yang S, Cho Y, Kang BB, Paek Y. Hypernel: A hardware-assisted framework for kernel protection without nested paging. In: Proc. of the IEEE 2018 55th ACM/ESDA/IEEE Design Automation Conf. 2018. 1−6.

[21] Ramos JR. TrustFrame, a software development framework for TrustZone-enabled hardware. https://fenix.tecnico.ulisboa.pt/downloadFile/1689244997256574/Extended_Abstract.pdf

[22] Winter J, Wiegele P, Pirker M, *et al*. A flexible software development and emulation framework for arm TrustZone. In: Proc. of the Int'l Conf. on Trusted Systems. Berlin, Heidelberg: Springer-Verlag, 2011. 1−15.

[23] Jiang Z, Audsley NC, Dong P. BlueVisor: A scalable real-time hardware hypervisor for many-core embedded systems. In: Proc. of the 2018 IEEE Real-time and Embedded Technology and Applications Symp. (RTAS). IEEE, 2018. 75−84.

[24] Jiang Z, Audsley N, Dong P. BlueIO: A scalable real-time hardware I/O virtualization system for many-core embedded systems. ACM Trans. on Embedded Computing Systems (TECS), 2019,18(3):Article 19.

[25] ARM. Trusted board boot requirements client (TBBR-CLIENT) Armv8-A. 2018. https://developer.arm.com/docs/den0006/latest

[26] Dong P, Burns A, Jiang Z, *et al*. TZDKS: A new TrustZone-based dual-criticality system with balanced performance. In: Proc. of the 2018 IEEE 24th Int'l Conf. on Embedded and Real-time Computing Systems and Applications (RTCSA). IEEE, 2018. 59−64.

[27] Zhao S, Zhang Q, Hu G, *et al*. Providing root of trust for ARM TrustZone using on-chip SRAM. In: Proc. of the 4th Int'l Workshop. ACM, 2014.

[28] Fan GN. The research of virtual TPM based on TrustZone [MS. Thesis]. Changsha: National University of Defense Technology, 2016 (in Chinese with English abstract).

附中文参考文献:

[1] 胡俊,沈昌祥,公备.可信计算 3.0 工程初步.网络与信息安全学报,2017.

[2] 田健生,詹静.基于 TPCM 的主动动态度量机制的研究与实现.信息网络安全,2016,(6):22−27.

[5] 郭颖,毛军捷,张翀斌,等.基于可信平台控制模块的主动度量方法.清华大学学报(自然科学版),2012,52(10):1465−1473.

[28] 范冠男.基于 TrustZone 的虚拟化 TPM 研究[硕士学位论文].长沙:国防科技大学,2016.

董攀(1978－),男,河南开封人,博士,副研究员,CCF 专业会员,主要研究领域为系统软件,系统安全,实时操作系统.

黄辰林(1976－),男,博士,副研究员,CCF 专业会员,主要研究领域为系统软件,信息安全.

丁滟(1977－),女,博士,副研究员,CCF 高级会员,主要研究领域为操作系统,系统安全,可信云计算.

范冠男(1992－),男,工程师,主要研究领域为分布计算,可信计算,信息安全.

江哲(1991－),男,博士,主要研究领域为实时系统,混合关键度系统,片上网络,虚拟化技术.