TrustZone技术的分析与研究

郑显义[1),2),3)] 李 文[2),3)] 孟 丹[1),2),3)]

[1)](信息内容安全技术国家工程实验室 北京 100093)

[2)](中国科学院信息工程研究所 北京 100093)

[3)](中国科学院大学 北京 100049)

摘 要 互联网时代的到来给嵌入式应用系统带来了前所未有的发展机遇,但是随之而来的网络应用安全问题也使得嵌入式应用系统面临着越来越严重的威胁,安全性已经成为嵌入式系统设计中一项极为重要的需求.为此,ARM公司提出了基于TrustZone技术的一套系统级安全解决方案,该技术是在尽量不影响系统的功耗、性能和面积的前提下通过硬件来实现安全环境与普通环境的隔离,而软件提供基本的安全服务和接口,由软硬件相结合而构建系统安全,也正因为这些特点而受到国内外研究者的广泛关注.文中重点分析了TrustZone技术提供的安全隔离系统基本架构、安全机制的实现方式及如何构建可信执行环境.在此基础上将该技术与其他提高嵌入式安全的技术作了分析对比,也进一步探讨了其优势与不足之处,并针对不足之处提出了可能的解决方案.最后,深入讨论了该技术在学术领域的相关研究工作和商业应用情况,同时结合当前嵌入式应用领域存在的安全问题展望了该技术的未来发展方向和应用需求.

关键词 嵌入式系统;TrustZone技术;系统安全;ARM;系统结构

中图法分类号 TP302 **DOI号** 10.11897/SP.J.1016.2016.01912

Analysis and Research on TrustZone Technology

ZHENG Xian-Yi[1),2),3)] LI Wen[2),3)] MENG Dan[1),2),3)]

[1)](*National Engineering Laboratory for Content Security Technologies, Beijing* 100093)

[2)](*China Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093)

[3)](*University of Chinese Academy of Sciences, Beijing* 100049)

Abstract The advent of the Internet era has brought the unprecedented development opportunities to the embedded application system, followed by the security issues of network applications which has led the embedded application system to be facing more and more serious threats. As a result, the security has become an extremely important requirement in the process of embedded system design. Therefore, ARM has proposed a set of system level security solutions based on TrustZone technology, which has implemented the isolation between the security environment and the normal environment by hardware and has also provided basic security services and interfaces by software. It has built the system security by combining hardware and software, however, it has no influence on performance, power consumption and area as far as possible. Due to those characteristic, the technology has gained wide attention of researchers from domestic and abroad. This paper has mainly analyzed the basic architecture of security isolation system provided by TrustZone technology, the way of security mechanism implement, and how to build the trusted execution environment. We have compared it with other technologies which can improve

收稿日期:2015-05-18;在线出版日期:2015-12-07.本课题得到国家"八六三"高技术研究发展计划项目基金(2012AA01A401)和国家"核高基"科技重大专项基金项目(2013ZX01029003-001)资助.**郑显义**,男,1986年生,博士研究生,主要研究方向为计算机体系结构、网络与系统安全.E-mail:zhengxianyi@iie.ac.cn.**李 文**(通信作者),男,1976年生,博士,高级工程师,主要研究方向为计算机体系结构和嵌入式系统.E-mail:liwen@iie.ac.cn.**孟 丹**,男,1965年生,博士,研究员,中国计算机学会(CCF)高级会员,主要研究领域为计算机体系结构、云计算、网络与系统安全.

embedded system security, discussed the advantages and disadvantages of the technology in further, and proposed the possible solutions aiming at the deficiency as well. Furthermore, we have discussed the related research work in the academic field and business applications of the technology. At the end, we have prospected the future development direction and application requirement of the technology combining with the current security issues in the field of the embedded applications.

Keywords　embedded system; TrustZone technology; system security; ARM; system architecture

1 引　言

随着网络应用在嵌入式系统应用中的日益深入，使得嵌入式应用和设备变得越来越复杂，另外很多开发者都基于此平台进行开发，使系统极易受到恶意攻击[1]，因此提供一套嵌入式系统安全解决方案的需求变得越发紧迫，也成为系统功能必备的要求. 在安全支付、数字版权管理(Digital Rights Management, DRM)[2]、企业服务和基于 Web 的服务等嵌入式应用中，安全性是消费者保护和商业价值内容保护的强制性要求. 如何避免设计上的缺陷变得越来越困难，不仅是因为系统复杂性给安全缺陷带来了机会，更是因为无法完全信任基于这个平台进行开发的众多开发者. 此外，由于互联网上存在各种各样的安全性威胁，当嵌入式设备连接到网络上时，它们难免会遇到极具威胁的网络攻击.

嵌入式产品在受到恶意攻击时能否保证安全是整个系统设计需要考虑的问题，设备中的硬件和软件必须相互配合才能保证在面对恶意攻击时能够实施行之有效的安全对策. 这是因为单纯依靠硬件的安全性解决方案是固定的，无法满足新的安全性要求，且设计时会增加硬件的开销，增加系统的功耗和复杂度. 而基于软件的安全方案则是通过数据加密或在操作系统中植入安全特性等方法实现系统安全，这样会大大增加系统的复杂性和成本，又因其数据交互的实时性和开放性，使之无法从根本上实现真正的系统安全[3]. 因此，有必要在嵌入式产品的软硬件设计中同时加入安全性措施，只有一个良好的系统硬件结构和适宜的安全软件设计才能确保安全产品不受外部恶意攻击，得以实现保护敏感数据的安全. 且增强系统安全的措施必须从整个系统设计一开始就着手实施，即从 CPU 内核和 SoC 基础架构设计开始时就需要考虑加入安全机制并将其集成到系统的整体设计中，使安全理念贯穿整个系统设计过程，而不是针对单个子系统提出一种安全机制.

嵌入式系统的安全性已经成为当前信息安全领域的关注热点，业界主流嵌入式处理器 IP 供应商 ARM 公司提出的 TrustZone 技术实现了一套系统级安全解决方案，它是在尽量不影响原有处理器设计的情况下提高了系统的安全性，引起了业界和学术界广泛关注. 它通过将保护措施集成到 ARM 处理器、总线架构和系统外设 IP 等措施来保证系统的安全性，并提供安全软件平台保证半导体厂商、原始设备制造商(OEM)和操作系统合作商可在一个共用的框架上扩展和开发自己的安全解决方案. 也正是通过硬件和软件组件的合理配合，该技术提供了一个具有高度安全性的系统架构，而对于内核的功耗、性能和面积的影响微乎其微. 此系统方法可以保护安全内存、加密块、键盘和显示器等外设，确保它们免遭软件攻击，并且能够建立一个隔离的可信执行环境(Trusted Execution Environment, TEE)[4]为安全敏感应用提供安全服务[5].

本文重点分析了 TrustZone 技术提供的安全隔离系统基本架构，安全机制的实现方式及如何构建可信执行环境，接着将该技术与其他提高嵌入式系统安全性技术进行了分析对比，也着重分析了其优势与不足，并进一步讨论了该技术的相关学术研究工作和商业应用情况，最后结合当前嵌入式应用领域存在的安全问题和该技术在安全方面的优势，展望了该技术的未来发展方向和应用需求.

2 TrustZone 安全架构概述

该部分将首先介绍 TrustZone 的软硬件架构及安全机制的实现方式，并在此基础上总结出该安全架构所面临的挑战.

2.1 TrustZone 硬件架构

TrustZone 硬件架构如图 1 所示，它将 CPU 内核隔离成安全和普通两个区域，即单个的物理处理

器包含了两个虚拟处理器核:安全处理器核和普通处理器核.这样单个处理器内核能够以时间片的方式安全有效地同时从普通区域和安全区域执行代码.这种虚拟化技术是在 CPU 设计时通过硬件扩展实现的,这些扩展可以保证安全内存和安全外设能够拒绝非安全事务的访问.因此,它们可以在正常操作系统中很好地隐藏和隔离自己,从而实现真正意义上的系统安全.这样,便无需使用专用安全处理器内核,从而节省了芯片面积和功耗,并且允许高性能安全软件与普通区域操作环境一起运行.并引入一个特殊的机制——监控模式,监控模式是管理安全与普通处理器状态切换的一个强大的安全网关.在大多数设计中,它的功能类似传统操作系统的上下文切换,确保切换时能安全的保存处理器切换前的环境,并且能够在切换后的环境正确的恢复系统运行.普通环境想要进入监控模式是严格被控制的,仅能通过以下的方式:中断、外部中断或直接调用 SMC(Secure Monitor Call)指令.而安全环境进入监控模式则更加灵活些,可以直接通过写程序状态寄存器(Current Program Status Register,CPSR),另外也可通过异常机制切换到普通环境.

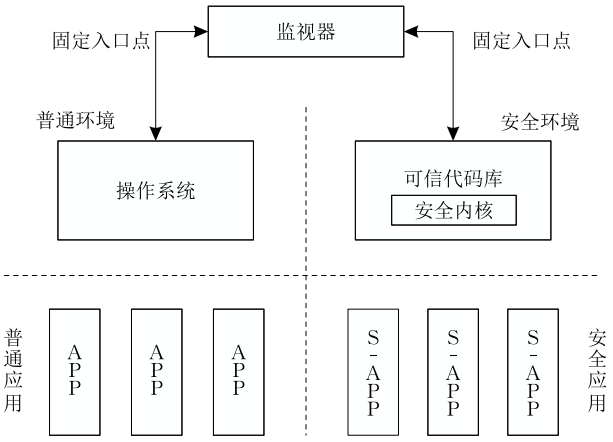

图 1 TrustZone 技术的系统架构

为了隔离所有 SoC 的软硬件资源,使它们分属于两个区域:用于安全子系统的安全区域和用于存储其他所有内容的普通区域,它在硬件架构上作了充分扩展.首先对内存进行了隔离,CPU 在安全环境(TEE)和普通环境(Rich Execution Environment,REE)[6]下执行进程时有各自独立的物理地址空间,REE 仅仅能访问自身对应的空间,而 TEE 有权限访问两个环境的物理地址空间.这样,软件在 CPU 处于 REE 下执行时,会被阻止查看或篡改 TEE 内存空间[7].而负责对物理内存进行安全区域划分的是地址空间控制器(TrustZone Address Space Controller,TZASC)和存储适配器(TrustZone Memory Adapter,TZMA),前者是 AXI 总线的一个主设备,用来将它从设备的地址空间划分为一系列内存区间,通过运行在安全环境的安全软件可以将这些区间配置为安全或非安全;后者则负责对片上静态内存 RAM 或片上 ROM 进行安全分区.并扩展了一些协处理器,协处理器可以通过扩展指令集或提供配置寄存器来扩展内核的功能.其中最重要的是 CP15 协处理器,它用来控制 Cache、可信加密模块(TCM)[8]和存储器管理.协处理器的某些寄存器在普通环境和安全环境各有一个,对这种寄存器的修改只能对所在的执行环境起作用.有的寄存器则会影响全局,比如控制对 Cache 进行锁定操作的寄存器,对这类寄存器必须严格控制,一般只对安全环境提供读写权限,而对普通环境提供只读权限[5].

中断是保证安全环境的重要一环,可以防止恶意软件通过进入中断向量的方法来对系统进行一系列的破坏.为此,对中断控制进行了扩展,普通环境和安全环境分别采用中断输入 IRQ 和 FIQ 作为中断源,因大多数操作系统都采用 IRQ 作为中断源,故采用 FIQ 作为安全中断源对普通环境操作系统的改动最少.如果中断发生在相应的执行环境,则不需要进行执行环境的切换.否则,由监控器来切换执行环境,且执行监控器代码时应该将中断关闭.在 CP15 协处理器中包含了一个只能被安全环境软件访问的控制寄存器,能够用来阻止普通环境软件修改 CPSR 的 F 位(屏蔽 FIQ)和 A 位(屏蔽外部中断),这样可以防止普通环境的恶意软件屏蔽安全环境的中断.而外设的安全主要由 AXI-to-APB 桥负责,比如中断控制器、计数器和用户 I/O 设备等.这使得其能够解决比仅仅提供一个安全的数据处理环境更加广泛的安全问题.该桥包含一个输入信号 TZPCDECPORT 决定外设是否配置为安全,该信号可以在 SoC 设计时静态地设置,也可以在程序运行时通过对 TrustZone 保护控制器(TrustZone Protection Controller,TZPC)进行编程动态地进行设置.而 TZPC 的安全状态是在 SoC 设计时确定的,它被设置为安全设备,只能被安全的软件环境使用.

上述硬件扩展是 TrustZone 系统架构的安全基础,可以看出它是在设计开始时就将安全措施集成到 SoC 中,并在尽量不影响原有处理器设计的情况下提高了安全性.类似硬件安全技术主要有 Stanford 的 XOM[9]和 MIT 提出的 AEGIS[10].其中 XOM 是

假定处理器内部单元能防御各种攻击且应用在 CPU 可信区而不是整个操作系统，它为阻止恶意软件攻击提供一个强有力的解决方案；AEGIS 是一个安全启动结构，为系统启动到应用提供多级验证. 相对这两种安全处理器技术，TrustZone 技术在硬件安全扩展具有较明显的优势，它能够为在其上运行的操作系统，比如 Palm OS、Linux、Symbian OS 和 Windows CE 等，提供一个系统范围的安全硬件架构基础，而对系统的功耗、性能和面积的影响很小，并已作为一个开放式安全架构和可信硬件平台受到业界的广泛认可.

2.2　TrustZone 软件架构

TrustZone 硬件架构扩展将安全性植入处理器中，这样为将安全性从普通操作系统（Rich OS，ROS）[7] 中分离出来提供了基础，即可以实现一个新的安全操作系统（Trusted OS，TOS）[11]，并加入监控代码区实现 ROS 和 TOS 之间的切换. TOS 和 ROS 同时运行在同一个物理 CPU 上，它们之间的交互限制在消息传递和共享内存传递数据. TOS 有独立的异常处理、中断处理、调度、应用程序、进程、线程、驱动程序和内存管理页表[7]. 监控代码区提供将两个系统衔接在一起的虚拟管理程序，并在两个系统过渡期间存储和恢复两个环境下寄存器的状态，并保证过渡到新环境下系统能够重新执行.

为了保证整个系统的安全，必须从系统引导启动开始就保证其安全性. 许多攻击者都会尝试在系统断电的时候进行攻击以便擦除或者修改存放在 FLASH 中的系统镜像. 因此，TrustZone 实行安全启动，大概流程是：设备上电复位后，一个安全引导程序从 SoC 的 ROM 中运行，该引导程序将首先进入 TEE 初始化阶段并启动 TOS，逐级核查 TOS 启动过程中的各个阶段的关键代码以保证 TOS 的完整性，也防止未授权或受恶意篡改的软件的运行[12]，随后运行 REE 的引导程序并启动 ROS，至此完成整个系统的安全引导过程. ARM 公司也定义了标准的应用程序接口（TrustZone API，TZAPI）[13]，这保证了软件和硬件开发者编写的应用程序可以被应用于不同安全平台的设备中，并允许客户端应用能够访问 TOS 以达到管理和使用安全服务的目的.

2.3　TrustZone 安全机制的实现方式

TrustZone 技术通过对 CPU 架构和内存子系统的硬件设计升级，引入安全区域的概念. NS（Non-Secure）位是其对系统的关键扩展，以指明当前系统是否处于安全状态. NS 位不仅影响 CPU 内核和内存子系统，还能影响片内外设的工作. Monitor 用来控制系统的安全状态和指令、数据的访问权限，通过修改 NS 位来实现安全状态和普通状态的切换. Monitor 不仅作为系统安全的网关，还负责保存当前的上下文状态. 并通过对内存子系统 Cache 和 MMU（Memory Management Unit）增加相应的控制逻辑来实现增强的内存管理. 其中，Cache 的每个 Tag 域都增加了一个 NS 位，这样，Cache 中的数据可以标记为安全和普通两类数据. 有两个虚拟的 MMU[14] 分别对应两个虚拟的处理器核. 页表项增加了一个 NS 位，相对应 TLB 的每个 Tag 域也增加了一个 NS 位，所有的 NS 位联合来进行动态验证，以确保仅得到授权的操作可以访问标记为安全的数据. 根据应用需求，该技术还可以将安全性扩展到系统其他层次的内存和外设上.

为了确保安全环境中的资源不能都被普通环境下的组件访问，保证两个环境具备强大的安全边界，相应对 AXI 总线上每个读写信道增加了额外的控制信号，分别是总线写事务控制信号（AWPROT）和总线读事务控制信号（ARPROT）. 这样，在 CPU 请求访问内存时，除了将内存地址发送到 AXI 总线上，还需要将 AWPROT 和 ARPROT 控制信号发送到总线上，以表明本次访存是安全事务还是非安全事务[5]. AXI 总线协议会将安全状态信息加载在两个读写信道控制信号 AWPROT 和 ARPROT 上，然后系统的地址译码器会根据 CPU 的安全状态使用这些信号来产生不同的地址映射. 比如，含有密钥的寄存器仅仅能被处于安全状态的 CPU 访问，实现访问操作是通过译码器将 AWPROT 或 ARPROT 置成低电平实现的. 如果 CPU 处于非安全状态试着访问这个密钥时，AWPROT 或 ARPROT 置成高电平，并且地址译码器将会产生访问失败，产生“外设不存在于这个地址”的错误[7]. 而 AXI-APB 桥则负责保护外设的安全性，普通环境不能够访问安全外设，这样就为外设安全筑起了强有力的安全壁垒. 将敏感数据放在安全环境中，并在安全处理器内核中运行软件，可确保敏感数据能够抵御各种恶意攻击，同时在硬件中隔离安全敏感外设，可确保系统能够抵御平常难以防护的潜在攻击，比如使用键盘或触摸屏输入密码.

TrustZone 技术所实现的运行环境使得安全性措施能应用于一个复杂嵌入式系统的很多层. 普通操作将完全地运行在 ROS 内，无需该技术的协助. 而为在 ROS 中实现安全性，该技术针对攻击方式提

供 3 种方式的完整性安全策略:首先,它会先从片内执行引导程序完成系统安全状态的配置才启动操作系统,只有通过安全验证的模块才允许被加载;其次,在系统运行期间,由 TrustZone 技术提供的安全代码区会处理普通代码区的安全请求,在处理之前把安全请求保存在共享内存中,当安全检测通过后请求会被处理;最后,一组受限的、可信的进程可以在远离 ROS 的私有空间内安全地执行.

3 TrustZone 构建的可信执行环境

GlobalPlatform(文后统一简称"GP")基于 TrustZone 技术制定了可信执行环境(TEE)的标准. TrustZone 是一种软硬件结合的系统范围的安全解决方案,通过硬件架构将资源隔离成安全环境与普通环境两个并行的执行环境,软件架构则提供能够支持完全可信的执行环境、安全敏感应用程序和安全服务的平台. 正是由于 TrustZone 技术从处理器内核设计时就通过硬件对资源进行了安全隔离,才有了实现 GP TEE 系统架构的基础. GP 制定的可信执行环境可作为一个独立的执行环境驻留在其所连接的支持 TrustZone 主处理器上的安全区域,以确保在可信执行环境中实现敏感数据的存储、处理和保护,其 TEE 系统架构如图 2 所示[15].

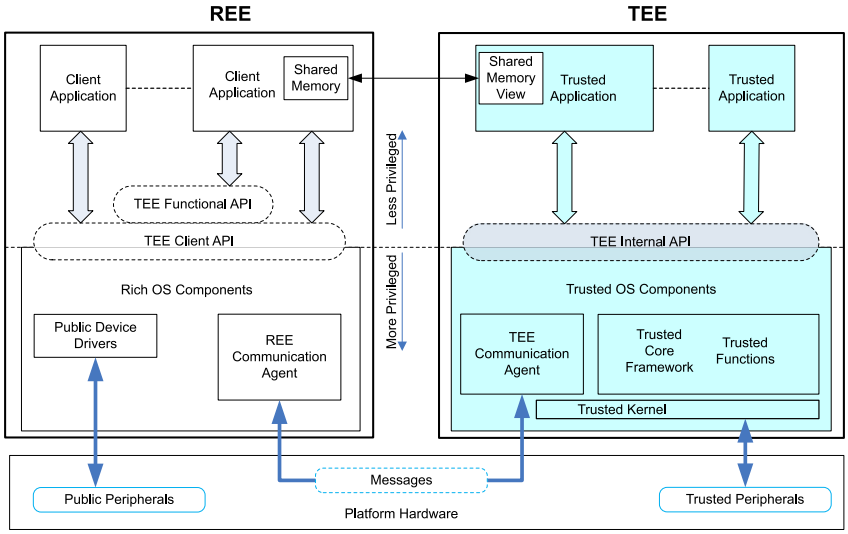

图 2 GlobalPlatform 可信执行环境系统架构

该系统架构中,TEE 是与设备上的 ROS 并行运行的独立执行环境,并且给 ROS 提供安全服务,TEE 内部由可信操作系统(TOS)和运行其上的应用程序,即可信应用(TA)组成. TOS 用来管理 TEE 的软硬件资源,并包含负责 REE 和 TEE 两种执行环境切换的监控器. TEE 所能访问的软硬件资源与 ROS 是分开的,TEE 提供 TA 的安全执行环境,同时也保护 TA 的资源和数据保密性、完整性和访问权限. TEE 中的每个 TA 是相互独立的,不经授权不能互相访问. TEE 自身在启动过程中必须要通过安全验证并且保证与 ROS 隔离. TEE 客户 API[16] 则是让运行在 ROS 中的客户端应用(CA)访问 TA 服务和数据的底层通信接口. TEE 功能 API 是对客户 API 的封装,封装了客户端与具体安全服务的通信协议,使得客户端能够以开发者熟悉的编程模式来访问安全服务,比如加密或可信存储[15]. TEE 内部 API[17] 提供给其 TA 的编程接口,内部 API 主要包括密钥管理、密码算法、安全存储、安全时钟资源和服务及扩展的可信 UI[18] 等 API. 可信 UI 是指在关键信息的显示和用户关键数据(如口令)输入时,屏幕显示和键盘灯等硬件资源完全由 TEE 控制和访问,而 ROS 中的软件则不能访问. REE 通讯代理提供了 CA 与 TA 之间通信的桥梁.

基于 TrustZone 系统安全平台构建的可信执行环境,一般都根据 GP TEE 系统架构标准. 电子科技大学的王熙友将 Android 系统作为 ROS,搭建了如图 3 所示的可信执行环境系统架构[19].

运行在安全状态的 TOS 管理 TEE 的软硬件资源,它在 Android 系统启动之前启动,启动后会对设备的安全属性进行配置,并分配安全存储空间. REE 的通信代理以 TrustZone 驱动的形式实现,TEE 的通信代理是一个守护进程,REE 与 TEE 的数据交换是通过共享内存进行的. 共享内存是在 REE 的内存区分配的内存,用于 REE 和 TEE 之间传递数据,主要是向 TEE 传递命令、参数及接受 TEE 返回的数据. Android 系统在 REE 下运行,不能访问存储

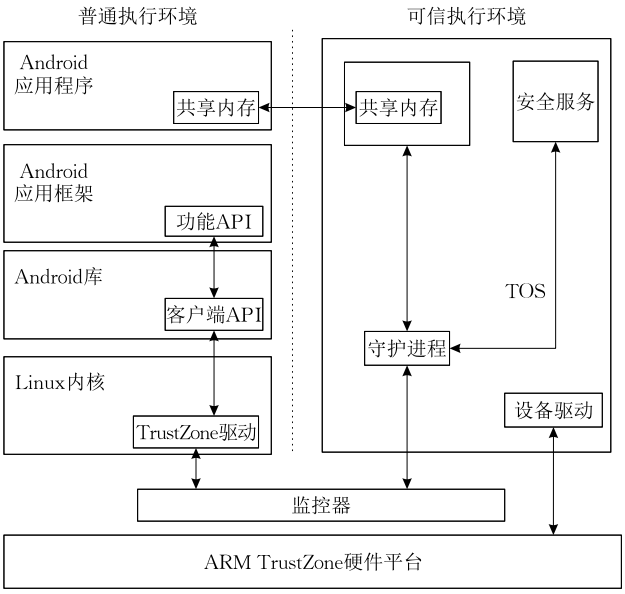

图 3　基于 TrustZone 的可信执行环境系统架构

在 TEE 中的敏感数据和配置为安全的外设，这样可以很好地保护用户敏感数据的安全，并可以保护安全内存、加密块、键盘和屏幕等外设，从而确保它们免遭恶意软件的攻击，因此基于该安全隔离技术构建的 TEE 系统架构能很好地解决 Android 系统目前存在的诸多安全问题.

上海交通大学利文浩等人基于 TrustZone 技术实现了具备 TEE 的可信内核 T6[20]，如图 4 所示，T6 采用 ARM 设备的硬件安全特性为移动操作系统提供较高的安全保障，ROS 支持 Linux 和 Android. 利用 T6 可以实现诸如移动安全支付、DRM 视频保护、ROS 内核防护及企业级 BYOD 保护等解决方案，即为用户设备程序的安全执行和隐私数据保护提供系统级的保护[21]. 在国外，格兰茨技术大学的 Andreas Fitzek 等人基于 TrustZone 技术也开发了一款 TOS 并命名为 ANDIX OS，它是一款支持多任务、非抢占式的操作系统[22].

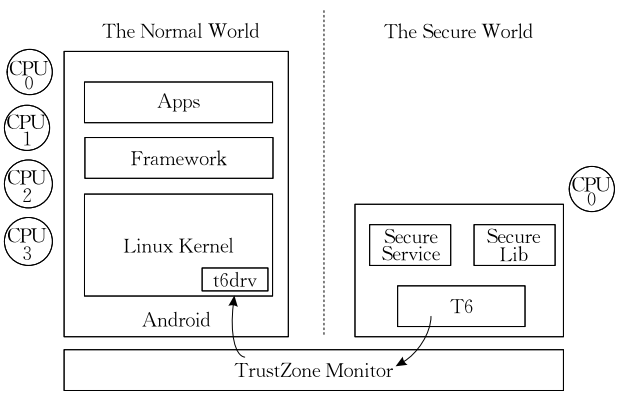

图 4　T6 系统架构

在商业界，Trustonic 公司是提供支持 TrustZone 安全扩展处理器的 TEE 供应商，能为智能移动设备提供可独立于 OS 的安全应用，它也执行 GP 标准. 捷德公司（Giesecke & Devrient）研发的 MobiCore 是一款基于该技术构建的可信内核，它能为程序执行和数据存储提供安全的执行环境. 其安全性也是建立在隔离原理基础上，即将 TEE 中的数据处理与 REE 中的数据处理隔离开. 其中，关键代码、加密密钥和敏感 I/O 等对安全敏感的应用运行在安全环境，而诸如用户 IO 等普通应用则运行在普通环境，这样做有效地保证了恶意软件对敏感数据实施的一些未授权访问. 欧版三星手机 GALAXY SIII 是第一款集成了 MobiCore 系统的智能机，它就是通过其在应用处理器中设置保护区的方式来实现安全运行并动态地下载对安全敏感的应用[23]. 另外，Open Virtualization（OV）提供开源的基于 TrustZone 的安全内核（SierraTEE）和监控器（SierraVisor），前者提供了一个隔离执行环境，使得安全敏感的代码与 ROS 系统完全隔离，它兼容 GP TEE API 规范并支持第三方可信代码载入执行. 但它不包含很多重要的功能，比如用户空间任务隔离、内核和用户空间隔离，多任务、动态载入应用、安全启动和 POSIX 兼容的库函数等[24].

4　TrustZone 技术的优势与不足

该部分通过将该技术与研究领域的其他提高系统安全的相关技术相比较，重点分析了 TrustZone 技术的优势与不足，并针对不足提出了可能的解决方案.

4.1　TrustZone 安全架构分析

TrustZone 技术为系统构造了一个安全的隔离运行环境，使其既能隔离不可信软件的潜在安全威胁，又能有效地运行被隔离软件，还能够监控其行为，从根本上解决了系统抵御不可信软件的安全威胁. 目前主流嵌入式系统中建立安全措施的方法主要有 3 种：SoC 设计外置硬件安全模块、SoC 设计内置硬件安全模块和软件虚拟化技术. 以下对 3 种解决方案的优劣势作详细分析，并与 TrustZone 进行比较分析.

系统设计中外置一个专用的安全硬件模块，比如手机中的 SIM 卡或机顶盒中设置访问条件的智能卡. 该方法可以将敏感数据保护在一个安全设计十分牢靠的物理设备中，也因为独立模块可以使用

完全独立的设计和制造流程,所以在设计和制造过程中可以充分考虑使用更先进的防篡改、物理安全技术和硅工艺.但它加重了 SoC 设计开销,增加系统的功耗和降低了处理器的性能,它也仅仅是提供了一个安全处理和安全存储的功能,当软件运行在安全硬件模块之外处理敏感数据时,很容易使数据受到外来的攻击.

第 2 种是系统设计中内置一个硬件安全模块,主要有两种形式:一是管理加密操作和密钥存储的硬件模块,二是内置在主处理器中的通用处理引擎.前者是通过使用内置安全硬件逻辑来阻止未授权的应用和进程对敏感资源的访问,相对第 1 种方法牺牲了部分硬件安全,却降低了设计成本,提高了系统性能而且便于集成.该方式也存在第 1 种方法中所述的安全作用范围受到限制,即在硬件安全作用之外,敏感数据也极易受到各种攻击.后者是为安全子系统提供专用的通用处理器,该方式与 TrustZone 技术的硬件安全解决方案比较类似,但也有不足之处.一是设计时需要一个单独的物理安全处理器会增加系统的功耗和硅面积,同时因为安全处理器和通用处理器之间通信时需要及时刷新共享内存中的数据,而共享内存通常是外置的,因此需要占用大量的执行时间.另外,资源分离需要在 SoC 设计中进行专门的硬件扩展从而大大增加了设计和测试工作,使得系统扩展变得非常困难.况且内置硬件安全模块的方法只能保证系统功能方面的安全,而完全没有考虑 SoC 在调试和测试模式下的系统安全,然而此时又极易受到攻击.如果关闭调试和测试模式,必然又使诊断软件问题变得非常困难[5].

第 3 种是软件虚拟化(VMM),它提供了隔离的执行环境,且 VMM 具有软件层中最高特权级,为安全带来多方面优势.VMM 管理的多个虚拟机独立运行在隔离环境中,不会被其他虚拟机的干扰和破坏.带 MMU 的任何处理器能实现虚拟化的解决方案,不需要额外的硬件来实现 VMM,安全敏感的软件可以移植到运行在 VMM 的安全环境中运行,而普通操作系统则在非安全环境中.但系统虚拟化忽略了与硬件攻击相关的攻击,比如调试和测试模式下遭受的攻击.此时要保证虚拟系统安全,必须禁用调试并且保证测试完全不可见,而这又使得软件开发和诊断软件的缺陷变得非常困难.另外,有些总线 Master,比如 DMA 引擎和 GPU 等,能绕过 VMM 提供的保护机制.同时,虚拟化技术因为需要在系统管理和资源分配方面做大量工作,因此它们自身也面对许多漏洞,它也因为需要模拟关键指令而增加了系统执行负载.

TrustZone 技术相比这 3 种提高嵌入式安全性的方案具有明显的优势,它是在尽量不影响系统的功耗、面积和性能的前提下提出的,它也是在整个系统设计中扩展了安全基础架构,包括系统的功能单元和调试的安全,而不是用一个只能保护敏感资源的专门安全硬件模块,它可以保护整个系统的安全,是一个彻底的安全解决方案.基于其硬件扩展架构有与之相配套的安全软件架构,能够保证安全启动,在受到黑客和木马等攻击时都会有相对应的安全应对机制,调试模式只有在得到授权以后才会开启.

4.2 TrustZone 与其他类似技术对比分析

目前,针对移动和嵌入式领域,TI 推出了类似 TrustZone 技术的安全隔离技术 M-Shield[25],如图 5 所示,它也是一种系统级的安全解决方案,紧密地结合了硬件和软件组件,能为各种利益相关者提供高级别的安全.M-Shield 技术为移动平台安全提供了安全的基础架构,可以进行高价值内容传送或存储.敏感应用的安全执行和数据的安全存储由硬件增强的安全环境进行保证,它定义了安全的 ROM 和 RAM 并嵌入了一个安全状态机(Secure State Machine,SSM),SSM 负责应用和管理在进入、执行和退出安全环境时的安全策略,且消除了芯片互连及 DMA 数据传输的脆弱性.安全的芯片互连允许安全环境和安全 DMA 通道对外设和内存进行访问,这样有效地保证了敏感数据在整个数据通道传送过程中的可信性.

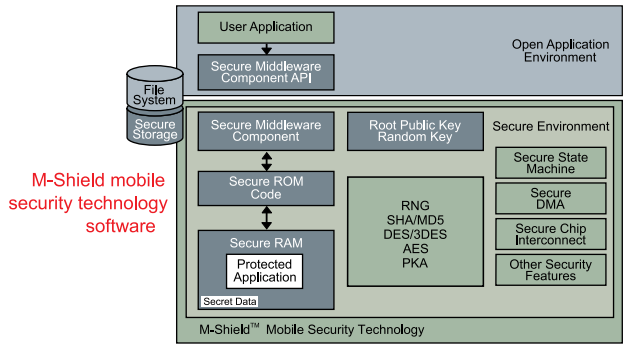

图 5 TI M-Shield 安全隔离的系统架构

为了补充该技术,TI 开发了一种安全中间组件,包括安全框架和基于 TrustZone 标准 API.它们提供了开放的移动安全框架,从而为使用 OMAP 与 OMAP-Vox 处理器的手机创造了可靠而灵活的应用环境.

而针对 PC 端,限于文章篇幅,这里作简要介绍.

Intel 和 AMD 就通过扩展 x86 指令集推出了 Late Launch 技术，分别为 TXT[26]（Trusted Execution Technology）和 SVM[27]（Secure Virtual Machine），它们允许软件模块运行在与 ROS 隔离的安全环境中，Flicker 和 TrustVisor 都是基于这些技术建立起来的可信系统. 他们指定 TPM 作为 Late Launch 的可信根，一旦 Late Launch 启动，运行在 CPU 隔离环境的软件代码被度量，度量值被存储在 TPM 中，程序运行结束后，TPM 的 AIK 密钥对此度量值进行签名以向外界证明运行在隔离环境中 APP 的完整性，但 TPM 硬件模块是针对 PC 端，并不适合对体积和功耗要求苛刻的嵌入式领域. 另外，Intel 近期推出了一项新的处理器技术 SGX[28]（Software Guard Extensions），它是一种旨在通过逆向沙箱的机制以提高软件安全性. 这种方法并不是试图识别并隔离平台上所有恶意软件，而是将合法软件封闭在一个地点，保护其不受恶意软件攻击，不论恶意软件有何种特权级别. 它能将安全应用依赖的 TCB（Trusted Computing Base）减小到仅包含 CPU 和安全应用本身，将不可信的复杂 OS 和虚拟机监控器 VMM 排除在安全边界外，然而目前市场上还没有支持 SGX 的 CPU 出现，而且模拟器也很难找到.

4.3　TrustZone 技术的优势与不足

TrustZone 技术是通过对硬件和软件部分的合理组合而设计的具有高度安全性的系统架构，而对于功耗，性能和面积影响微乎其微. 因此，该技术在提高嵌入式系统安全方面拥有大量的技术和商业上的优势，主要分为以下几个方面进行阐述. 首先，它可以为片上的保密数据提供安全的隔离环境，而这种处理方式也是目前保密的最佳途径. 例如，如果想用 SoC 上的一个 CPU 来处理 SIM 卡中的密钥，必须确保在 SoC 环境中有个完全安全的区域，一个不怎么安全的 OS 是不可以做这些操作的[3]. 其次，性能一直是某些保密系统中难以克服的问题，特别是在片上处理器和片外存储器之间需要频繁传递信息时，这些信息必须经过加密. 此时 TrustZone 便可发挥作用，因为它对整个存储空间都可以保证完全的总线带宽，而在其安全缓冲区数据却可以以明文的形式存储从而实现快速访问. 加密过的数据则可以以普通方式存放在 FLASH 存储器中，这样可以使用一些便宜的，容量大的，灵活的存储方式[3]. 另外，TrustZone 系统架构是软硬件的合理组合，即使在 SoC 设计完成后，它依然可以保证用户能够灵活地定制和升级保密系统. 最后，TrustZone 在嵌入式系统中定义了一个安全的隔离环境，该独立环境中包含一些直接的外设通道，用户界面，SIM 卡，智能卡及音频输出等. 对于非保密部分，TrustZone 可以通过完整性检查机制为 SoC 器件中的所有部分提供安全保护. 例如，解码后的 DRM 音频数据被传送到非保密区域中时可以通过操作系统有关部件的完整性检测来受到保护[3].

虽然 TrustZone 技术是提高嵌入式系统安全性比较行之有效的系统级安全解决方案，体现了很多的安全特性：平台的识别和认证，密钥管理，底层加密技术，I/O 访问控制，安全数据存储，智能卡控制及代码/完整性核查等[3]. 然而，它也不是万能的，自身也存在许多不足，主要体现在以下几个方面. 首先，它只能很好地防御各种软件攻击，难以防止物理攻击，比如物理篡改设备的主存. 另外，虽然它可以通过度量机制保证安全隔离内核代码的安全性，也会定期对 ROS 内核做完整性检查，但此时攻击已经发生，因此无法真正防御对系统的恶意攻击. 其次，它仅仅提供了一个隔离的执行环境，而没有向用户或远程者证明这个环境是可信的. 最后，为系统平台提供一个可靠的可信根是整个系统安全的基石，而目前该技术是通过在片上系统固化设备密钥作为可信根，这种方法会存在密钥更新困难，一旦泄露会导致整个平台无法使用的弊端. 且这种方式需要将设备密钥长期存储在设备上，其安全很难保证，比如如何防御旁道攻击、故障攻击和逆向工程等类型的攻击. 因此，如何在不需要在 TrustZone 现有硬件安全基础架构内增加硬件的前提下提供一个可同时防御物理攻击和软件攻击的信任根，保证系统从设备上电到运行都在可信的执行环境下，也必须要重点考虑. 当然，面对嵌入式领域日益严峻的安全需求，如何保证基于 TrustZone 安全隔离的执行环境在提供各种敏感数据处理和安全服务的同时保证其 TCB 尽量小，使 TOS 安全性得到足够保障也显得非常关键.

虽然可信计算组织（Trusted Computing Group，TCG）结合硬件和软件，为实现更安全的计算环境，发布了适合 PC 平台的可信平台模块（TPM）[29]，Intel 和 AMD 都指定 TPM 作为 Late Launch 的可信根，但 TPM 硬件模块并不适合对面积和功耗要求苛刻的嵌入式领域. 随后 TCG 针对嵌入式和移动应用又发布了移动可信模块（Mobile Trusted Module，

MTM),并引入可信启动,但遗憾的是 MTM 是功能而非硬件实现[30].有文献研究也对 MTM 的性能进行了重新评估,认定 MTM 会增加系统的功耗并降低加密功能的性能[31].因此,目前将 MTM 作为嵌入式系统的可信根还不是一种理想的选择.而基于 SRAM PUF(Physical Unclonable Functions)技术提取密钥作为可信根的方式会是一种比较行之有效的方法,它直接集成设备 SRAM 的物理特性,因此不需要额外的硬件资源,可以降低设备的 TCB.另外,SRAM PUF 技术作为密钥提取和安全存储,可以抵御各种软硬件攻击,比如抵御逆向工程和有效防止克隆等,从而在一定程度上解决 TrustZone 难以防物理攻击的弊端.另外,它也可以用于构造随机数发生器,免去硬件构造带来的成本、性能和功耗的影响.而针对 TrustZone 技术隔离的 TEE 没有向用户或远程者提供证明运行在 TEE 的软件没有被恶意代码篡改的证据,可以将该技术结合软件 MTM 提供的远程证明功能实现.TrustZone 可以保证软件 MTM 运行在一个安全的隔离环境中,而软件 MTM 可以远程证明 TOS 是可信的.除此之外,它还可以为系统平台提供丰富的可信计算功能,比如安全存储、身份认证、平台安全保护等.当然,在考虑系统安全的同时,必须充分考虑应用需求,针对具体应用场景进行设计以保证 TOS 的 TCB 足够小,实现真正意义上的系统安全.

5 基于 TrustZone 的相关研究工作

针对嵌入式系统及其应用的安全问题日益突出的现状,业界成功推出了基于 TrustZone 技术的安全解决方案,学术界也围绕该技术展开相关研究.通过归纳总结,其相关研究工作主要分为以下几个方面.

5.1 基于 TrustZone 构建安全平台

TrustZone 技术提供了一个安全的基础架构,开发者可利用此技术构建自己的安全平台以满足各种安全需求,比如安全支付、指纹识别、DRM 等.Samsung 公司基于开源的 Android 系统和 TrustZone 技术推出了全新全方位的移动安全平台 KNOX,它是利用安卓安全增强(Android SE)[32-33]执行强制访问控制策略来隔离平台内的应用程序和数据,从而为平台和应用都提供安全保障[34],其系统框架如图 6 所示.

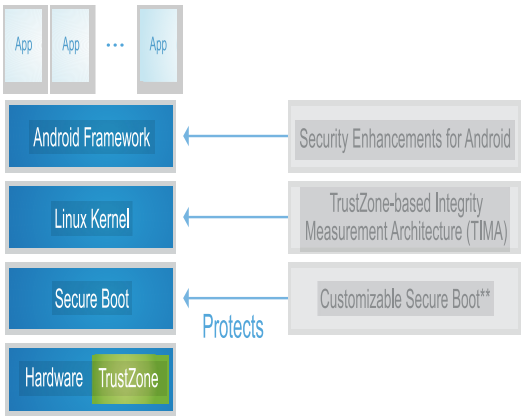

图 6 KNOX 系统安全架构

Android SE 安全机制需要保证操作系统内核完整,否则会失效.KNOX 系统架构中基于 TrustZone 的完整性测量结构(TrustZone-based Integrity Measurement Architecture,TIMA)就是为了关闭这个漏洞,它使用 TrustZone 硬件架构有效地将内存和 CPU 资源划为安全区和普通区,其中 TIMA 运行在安全区,不能被禁用,而 Linux 内核的 Android SE 运行在普通区.TIMA 实时对 Linux 内核进行连续完整性监测,当 TIMA 检测到内核的完整性受到攻击时,它会通过移动设备管理(MDM)通知企业 IT,企业 IT 能采取相应的策略来保护内核的完整性.这样,安全启动和 Android SE 及 TIMA 形成防御恶意攻击内核和核心辅助进程的第一道安全防线.

AMD 公司提供内置于 AMD APU 中的专门平台安全处理器(Platform Security coProcessor,PSP),它是 AMD64 核的集成协处理器,如图 7 所示,它通过利用 TrustZone 技术将 CPU 分为两个虚拟区域来打造安全环境,敏感任务运行在 PSP 上,即在安全区域运行,而其他任务则在普通模式下运行.这样能够确保安全存储以及处理敏感数据和可信赖的应用程序,并能很好地保护关键资源的完整性和机密性[35].

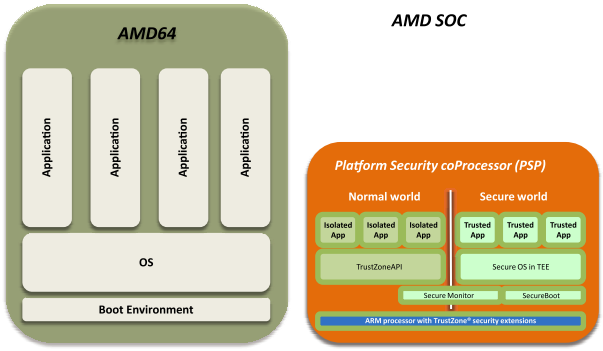

图 7 AMD PSP 设计框架

2014 年 1 月 AMD 推出首款 ARM 架构服务器处理器 Opteron A1100，它是一款完整的 SoC，而不是 CPU，具备完整的功能设计. Opteron A1100 基于 ARM 的 Seattle 架构核心设计，芯片内核架构为 64 位 Cortex-A57，四核心设计，芯片的默认频率为 2 GHz，功耗低于 20 W，拥有 4 MB 的二级缓存和 8 MB 的三级缓存，支持双 DDR3 或 DDR4 存储通道和双万兆以太网接口，它是第一个支持 TrustZone 安全模块的服务器处理器，充分利用了该技术在安全方面的优势，为提高此类服务器的安全起到了非常重要的作用.

Apple 公司定制了一个高度优化过的 TrustZone 安全框架，并以此为基础推出了 Secure Enclave 模块，能够很好地解决如何加密、存储和保护用户指纹类这些极重要的生物学信息，并负责验证来自 Touch ID 的指纹数据，如果数据匹配则启动访问或购买. Secure Enclave 模块是内置在 Apple A7 芯片中的协处理器，具有独立于主处理器的安全启动和软件更新机制. A7 通过串行外围接口总线（SPI）与 Touch ID 通信来获取指纹数据但不能读取数据的内容并将数据传递给 Secure Enclave，由其负责数据的加密操作和完整性保护[36].

业界推出的安全平台都是基于 TrustZone 提供的硬件安全隔离环境进行敏感信息处理和安全存储及设计安全服务来服务于应用需求. 以上三星的 TIMA 完成 Linux 内核的完整性核查，AMD 的 PSP 负责应用敏感信息的处理和 Apple 的 Secure Enclave 负责处理指纹类生物信息的处理都是基于此思想. 由于敏感信息处理和安全存储及安全服务都在一个受到 TrustZone 硬件隔离的保护且与 ROS 完全隔离的环境中，其安全性能够得到足够保证.

5.2　基于 TrustZone 构建安全系统环境

系统功能越来越复杂，代码量越来越大，无法避免存在各种漏洞，黑客可以利用这些漏洞实施攻击从而获取系统的敏感信息. 利用 TrustZone 的硬件安全隔离优势可以保障 ROS 系统的安全. 文献[37]提出了基于该技术的实时内核保护（TZ-RKP）机制，如图 8 所示，ROS 内核中控制指令和页表更新功能会放到 TOS 中审核后执行而不允许 ROS 对其直接修改. 它主要强制对内核中某些特权系统功能放到安全环境中进行核查和授权后才允许执行，从而有效的阻止修改和添加内核文件的攻击. 但此文针对诱骗内核修改自身数据类攻击没有在 TZ-KRP 内实现相应的检测机制和处理机制，而这类攻击能劫持内核控制流使其严重损坏.

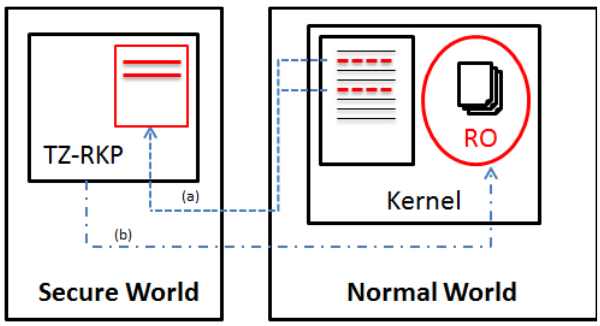

图 8　TZ-RKP 设计框图

文献[38]针对嵌入式系统提出了一种高安全系统原语平台，主要将 TrustZone、TPM 和可编程安全控制逻辑（PSCL）结合在一起，该平台改善了嵌入式系统性能并提高了系统安全性. 其中，PSCL 主要包括以下组件：安全有限状态机（SFSM）、可编程安全数据通路（PSD）及可编程安全处理模块（PSPM）. TPM 仅仅与安全内核交互，主要提供安全存储、内核完整性度量和系统安全完整性报告. 当系统运行可信应用时会先配置 SFSM，SFSM 将核查需要使用什么外设，并基于此 SFSM 定义 PSD 告知安全 CPU，安全 CPU 选择合适的参数并将其写入配置寄存器，同时 SFSM 基于这些参数构建 PSPM. 当 PSD 和 PSPM 被构建，SFSM 会继续通过安全 CPU 监测系统当前所处的状态是否安全，一旦系统不安全，安全 CPU 会复位 SFSM 并会刷新 PSD 和 PSPM. 但此文对该设计的安全策略、性能及安全 CPU 与 SFSM 之间的安全通讯协议没有作深入分析. 文献[39]基于 TrustZone 和安全 Linux 系统为嵌入式系统构建了一个安全增强框架，该框架由多策略访问控制机制和一种安全增强方法组成. 其中，多策略访问控制机制通过利用 DTE（Domain and Type Enforcement）模型和改善的 BLP（Bell-La Padula）模型实现；而安全增强方法通过雇佣 Linux 安全模块（LSM）框架从而为系统提供强有力的防护. 普通环境利用安全 Linux 提供的 BLP 和 DTE 策略避免恶意攻击，保证系统的完整性和机密性，而安全应用通过调用基于 TrustZone 隔离的安全环境内的安全服务进行处理. 该原型设计能够为开放的嵌入式系统和各种应用提供安全的执行环境，但本文没有结合具体的应用场景，也没有具体实现的安全服务. 文献[40]针对抗恶意攻击工具为了监测到最新的恶意攻击需要持续更新提出了基于 TrustZone

的内存获取机制 TrustDump,它能够获取 ROS 中 RAM 和 CPU 寄存器的数据.即使 ROS 受到损坏,TrustDump 也能保证自身的安全,该方法很好地实现了恶意进程检测和内核完整性度量,从而保障了系统的安全.然而,此文对获取的数据只做了相对简单的在线分析,且没有针对检测到的攻击实现的具体处理方案.

5.3 基于 TrustZone 构建可信计算环境

TrustZone 的隔离环境需要一个可信计算环境为系统平台提供丰富的可信计算功能.基于该技术构建 MTM 是一种常见的构建可信移动平台的方法,为 MTM 的运行提供安全保障.文献[41]融合 TCG 的可信计算概念和该技术建立起一个开源的基于 Linux 嵌入式可信计算平台.具体是在 TrustZone 的安全区域构建了一个虚拟化框架,并基于此基础上设计了一个可信移动平台原型,并实现了安全启动,这个原型实现了纯软件的 MTM,而不需要增加额外的硬件.该文也证实了利用硬件安全机制实现嵌入式可信计算软件平台的可行性.但此文基于开源操作系统 Linux 作为基础安全部件构建 MTM 软件,存在 TCB 过大的弊端,安全性得不到足够保证.文献[42]分别讨论了基于 TrustZone 和具体安全组件(如智能卡 JavaCard)提供的隔离环境构建软件 MTM,分析得出基于 TrustZone 能提供类似硬件 MTM 的保护能力,基于智能卡实现的软件 MTM 能匹配硬件 TPM 的保护机制,但本文对这两种实现方式的区别没有具体的论述,也因为当时没有具体公开有效的 MTM 测试套件而不能确认他们的实现是否完全符合 TCG 规范.文献[43]提出一种便携式的可信计算模块 TEEM,它能为各种计算平台,比如桌面机和移动设备,提供各种可信计算功能,TEEM 设计成一个 TPM 服务运行在 TrustZone 的安全区.然而,此文的实现没有将 TEEM 与 ROS 隔离,事实上 TEEM 运行在整个 Linux OS 上,这导致 TCB 非常大,且没有在开发板上进行验证,也没有结合具体可信应用场景.

5.4 基于 TrustZone 构建安全服务

嵌入式系统日益丰富的需求,各种需求的应用应运而生,其中对安全敏感的应用会涉及大量用户隐私信息,基于 TrustZone 提供的安全隔离环境构建安全服务来为这类应用提供敏感数据处理以防止恶意攻击是一种比较可行的方案.文献[44]利用该技术实现了一个全新的移动在线购票系统,在线交易过程中的敏感数据会保存在 TrustZone 硬件安全区,从而很好地保证了交易过程中敏感数据的安全.客户购票在线支付完成后,商家会发送一个加密的电子票给客户,然后客户会给商家发送一个加密密钥,商家收到该密钥会反馈一个加密密钥到客户移动终端的 TrustZone 安全区,这样客户就能通过该反馈密钥打开并查看电子票详情.因为整个操作过程敏感数据都有效地被保护在独立的安全执行环境下,从而确保交易过程中电子票敏感信息的安全.文献[30]针对目前支付系统中没有较好地考虑隐私保护的问题,于是结合当前的支付机制提出了基于 TrustZone 的隐私保护平台,它适用于在线远程支付时(如 NFC),需要隐私保护的各种应用程序,如图 9 所示.具体实现是:在线支付时支付应用可以通过平台内部 TrustZone API 机制与可信应用通信,将支付时的隐私等敏感数据存储在 TrustZone 隔离的安全环境中,同时这些数据在流动时会受到安全监控器的监控,防止遭到恶意攻击导致的隐私泄露.文献[45]分析了目前典型云存储服务 Dropbox 存在的弊端:所有加密密钥由软件管理,无法对客户端软件的完整性进行认证及基于 ID 和密码的用户登录认证机制容易受到攻击.为了克服这些弊端而提出了一种安全的数据访问控制方法 DFClound,它依赖设计在安全隔离环境的 TPM 服务管理所有加密密钥并在合法用户之间定义了密钥共享协议.该原型设计实现是使用 ARM Fastmodel 软件模拟 ARM Cotex-A15 核,并利用 Open Virtualization (OV)[24]提供基于 TrustZone 的可信执行环境,它给数据的云存储和访问提供了一个安全环境.但此文提出的原型设计没有在具体开发板上实现,且性能负载非常大.

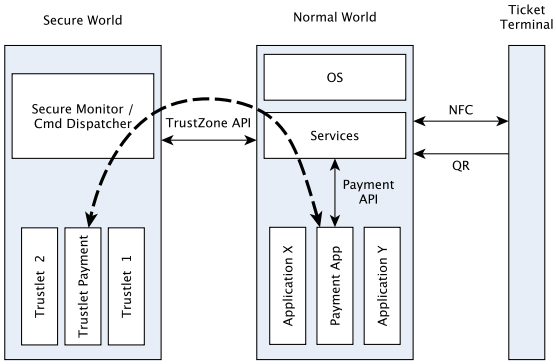

图 9 基于 TrustZone 的安全支付平台

文献[46]通过将 Symbian DRM 和 TrustZone 技术相结合了实现移动设备上的数据版权保护.文献[47]针对 TrustZone 技术不能阻止离线回滚攻

击，即在闪存中用先前的值替代当前的值，提出了一种能够防御此类攻击的解决方案. 该方案主要是依赖 TrustZone 技术能够为安全存储提供的一块 FLASH 逻辑分区和为可信应用软件提供的一个安全的执行环境实现的. 这些文献研究的思路都是将敏感应用涉及的敏感数据处理过程放置在由 TrustZone 提供的硬件隔离环境构建的安全服务来完成，这样可以有效地保证敏感数据的处理是在一个安全的环境中完成的.

5.5　基于 TrustZone 构建安全启动

要实现系统安全，安全启动是基石，只有这样才能保证系统的运行环境真正可信. 因此，基于 TrustZone 构建安全启动的研究也非常多. 文献[48]基于 SRAM 的物理不可克隆特性(SRAM PUF)[49]为 TrustZone 平台重构可信根以保证启动安全，如图 10 所示. 具体实现是：首先在片上 SRAM 内实现构建块，主要负责从 SRAM 初始响应提取原始种子(PS)和随机数种子(TRS)，其中 PS 用于产生唯一的设备密钥，TRS 用于为 TOS 建立一个安全随机数产生器(RNG)，它们都是建立可信根的基础，同时构建块也提供 TOS 和安全服务的安全启动. 随后，利用设备密钥提供密封/解密原语服务于 TEE 中的安全服务，并在 TEE 环境集成纯软件的 TPM 服务为 ROS 提供丰富的 TPM 服务，这样有效的保证了 ROS 免受软件袭击，它作为 ROS 的可信根. 由此，在系统启动到正常运行形成了一个可信链，而且 ROS 的应用可以利用 TPM 服务将信任链扩展到应用层，整体设计有效地保护了系统的安全性. 此文基于 TrustZone 提供的内存隔离机制将信任根与 ROS 完全隔离，避免了来自 ROS 的软件攻击，然而，其主内存没有实现而在 SoC 外，所以该设计的信任根不能抵抗直接攻击硬件平台的物理攻击.

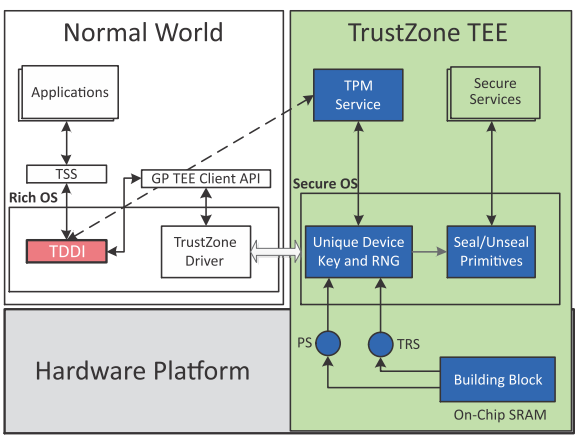

图 10　TrustZone 可信根设计框架

随着无线传感网络应用越来越广泛，比如智能家居系统、运输系统及军事领域等，其安全问题变得日益突出，最重要的就是设备启动阶段所受到的物理攻击. 基于此，文献[50]分析了当前大多数无线传感节点依靠软件来保护系统安全，但随着它在军事和健康监测等敏感领域的应用，基于软件的安全防护已经捉襟见忖. 于是，基于 TrustZone 提供的硬件隔离安全和安全内存配置等特性提出了无线传感节点的安全启动系统，然而，此文没有具体实现也没有进行安全性分析. 文献[51]分析了安全启动和可信启动存在的问题，前者定制于特定设备，用户不能自由选择软件；后者缺少运行时验证的机制. 并在此基础上基于 TrustZone 技术提供安全硬件隔离提出了二次启动验证架构，它很好地解决了上述两者的缺点. 具体是：第 1 阶段启动时验证引导程序和 OS 镜像并进行登记；第 2 阶段运行的应用能核查启动痕迹并能验证运行软件是否满足安全条件.

5.6　基于 TrustZone 构建虚拟化平台

系统虚拟化(VMM/hypervisor)[52-54]为各类应用能够提供很好的隔离执行环境，然而，它自身存在许多的问题，本文前面章节已做深入分析. TrustZone 从广义角度也是一种虚拟化技术，相对于 VMM，它提供了硬件隔离和内存保护机制. 因此，基于 TrustZone 可以增强现有的软件虚拟化技术的安全性，克服其本身存在的不足. 文献[55]针对目前虚拟化技术中因为需要模拟关键指令而给带来较重的负载，基于 TrustZone 的普通环境和安全环境拥有各自的特权模式和用户模式的特点提出了 ViMoExpress，它是嵌入式系统的轻量级虚拟化解决方案. 两个系统分别运行在 TrustZone 隔离的普通区和安全区，ViMoExpress 运行在 Monitor 模式，这样 ViMoExpress 产生很少的负载，该设计在单核 ARM 处理器上实现的 ViMoExpress 加速了两个系统，ViMoExpress 仅仅的负载是两个系统切换的中断时间，且添加的代码量非常小. 然而，此文没有针对设计进行具体的性能分析和安全性分析. 文献[56]基于 TrustZone 安全扩展实现了非对称虚拟化层，它支持在单个处理器上同时运行 RTOS 和 GPOS，该实现不需要修改 GPOS 且因不需要特权指令模拟使执行负载很小. 此文也强调 VMM 代码需要进一步优化，考虑为 GPOS 设计专门的设备驱动的方式扩展 VMM 以使 RTOS 和 GPOS 之间支持多路通讯通道. 文献[57]在没有修改通用操作系

统(GPOS)的前提下基于 TrustZone 虚拟化提出一个软件架构 SafeG 作为 Monitor,如图 11 所示.

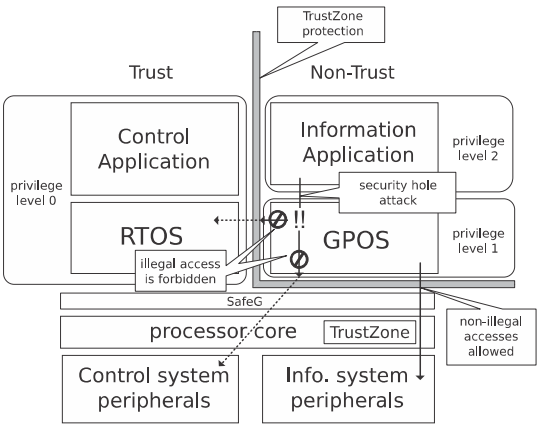

图 11 基于 TrustZone 的 VMM 设计框架

该框架实现了 GPOS 和实时系统(RTOS)同时运行于单个处理器上,其中 RTOS 用于处理实时性要求较高的任务,而 GPOS 处理用户普通任务. SafeG 对外设进行了隔离,使其能够按照需求将硬件外设实时分配给需要的操作系统,同时减少了 RTOS 的隔离负载并提高了其运行可靠性. 正是由于这些特点,该实现很适合于汽车导航系统、移动电话或机床等应用场景. 然而,此文也提到要增加 TrustZone 作为一个虚拟化硬件的一个机制是可信区和普通区必须有独立的 Caches,而当前 Caches 没有这样设计,这样普通区刷新 Caches 时会影响可信区的性能. 另外,从普通区切换到可信区有 38 个寄存器需要持续保存和恢复,引入保存和恢复这些寄存器的指令能改善它的性能. 且此文设计不支持 TrustZone 扩展的多核处理器当中.

5.7 小 结

基于 TrustZone 技术的商业应用和学术研究已经涉及到嵌入式系统的方方面面,为嵌入式系统的系统安全提供了可靠的保证. 基于其的研究和开发工作,无论是商业应用还是学术研究对解决目前嵌入式系统存在的安全问题是至关重要的,可以基于该技术提供的系统级安全框架开发满足特定需求的安全平台及设计全新的安全策略.

6 TrustZone 研究和应用需求展望

目前基于 TrustZone 技术在系统安全中的应用研究工作还处于初始阶段,在上述多个方面还存在许多挑战未能解决. 展望未来,针对目前该技术在系统安全方面的应用所面临的问题,以下 3 个方面的应用仍然需要不断探索和研究.

6.1 利用 TrustZone 技术作系统级防护

ROS 因代码量大、功能丰富和开发应用环境复杂等原因而无法避免存在各种漏洞,其安全性问题已经普遍存在. 如何利用 TrustZone 技术的硬件安全隔离优势实现 ROS 的系统级防护并保障其敏感应用的安全,已经成为解决此类安全问题的比较行之有效的方法. 以开放的 Android 系统为例,它给用户带来了许多便利,比如允许用户下载来自不同开发者提供的应用和服务,但这也使得一些对安全敏感的应用很难保证其在执行时的安全[58]. 尽管 Android 系统通过集成 Linux 2.6 内核的安全机制实现系统安全,又通过自身的 permission 机制实现数据安全[59]. 但面对复杂的安全环境,依靠系统本身的安全机制是远远不够的,这大大降低了该系统的可用性. 因此,需要为该系统提供一个硬件隔离出来的独立可信的执行环境,从系统的底层检测和控制系统的运行行为,才能真正意义上实现整个系统的安全. 而 TrustZone 技术不仅可以在尽量不影响系统性能的前提下利用其提供的安全隔离环境来处理 Android 系统中对安全敏感的应用,为这类应用提供各种安全服务,比如安全支付、安全输入和安全显示等,还可以利用其对该系统进行防护,比如采用系统运行内核度量机制、敏感操作核查及敏感数据加密等,以增强 ROS 在复杂嵌入式环境下的抵抗恶意攻击能力.

6.2 TrustZone 技术作用于封闭式系统

TrustZone 技术的安全保护机制并非只可用于使用操作系统的开放式系统中,也适用于深层嵌入的或封闭的系统当中,在这些系统当中也能得到充分应用,如何基于该技术保证封闭式系统的安全也变得非常关键. 比如目前汽车系统正发展基于控制软件的集成扩展软件,软件集成增加了软件的复杂度,这可能导致系统故障而威胁汽车的安全. 为了解决这个问题,文献[60]就提出扩展软件应与控制软件必须真正隔离起来,于是基于 TrustZone 安全架构设计了一个安全的汽车软件平台,该平台实现了安全的设备访问,它限制直接访问扩展软件并支持多核处理器. 而该技术的应用场景也远非是汽车系统等封闭式系统当中,任何嵌入式应用都能受益于其提供的系统级安全架构,比如消费娱乐系统、硬盘驱动器等,它能为系统各个环节提供安全增强. 然

而，虽然该技术被设计用于在复杂开放的系统中提供更高级别的安全性，而有严格安全性要求的简单系统也能受益于该技术. 也正是由于其能为各种系统和系统中各个环节的安全需求提供支持的这些特点，基于其在系统安全中的研究和应用还有许多工作要做.

6.3　TrustZone 应用于其他系统架构

TrustZone 安全架构并非局限于 ARM 架构的处理器，x86、MIPS 等架构也可以引入该安全技术，可以利用其安全优势来支持这些架构设备的系统级安全. 如何实现该技术与其他系统架构结合，充分利用两者各自的优势保障系统安全也变得势在必行. AMD 就将 x86 架构与该技术融合在一起推出了其相应的应用产品，其 2014 年 4 月推出了第三代主流与低功耗 APU Beema/Mullins，Beema 偏主流，Mullins 则主打超低功耗，其系统架构如图 12 所示.

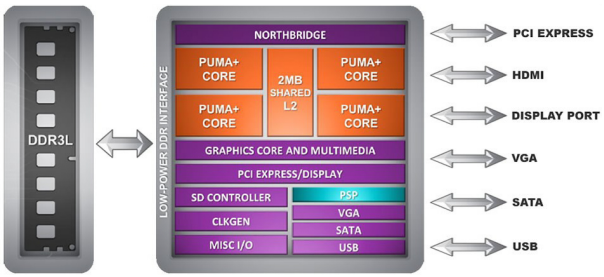

图 12　AMD Beema/Mullins APU 系统架构

该架构融入了 Cortex-A5 架构的 TrustZone 技术，也是第一台匹配 ARM 硬件加密安全模块的 x86 处理器，AMD 称其之为平台安全处理器(PSP)，PSP 的使用原理在本文之前章节也已作了详细分析. 因此，不同系统架构结合该安全技术在安全隔离方面的优势来保障系统安全，虽然不同架构技术融合过程中会存在很多技术难题，但深究其的研究对增强其他系统架构的系统安全有着非常重要的意义.

此外，该技术在开源软件方面的研究工作也是广泛且切实可行的. 文献[61]中 Winter 认为目前基于 TrustZone 系统开源软件开发工作相对较少的原因是由于缺乏支持该技术的低成本的开发板，于是分享了使用支持 TrustZone 廉价的开发板进行开发工作的经验，并且证实这项成果适合学术研究和教学. 他也在另一文献[62]中证实了使用开源模拟处理器软件实现该安全技术的软件开发可行性，如支持 TrustZone 的 QEMU 模拟器.

总之，随着嵌入式系统的发展及应用需求不断扩展，安全问题已经迫在眉睫，基于该技术的开发研究工作与商业应用会深入到嵌入式系统安全领域的方方面面，而在其他封闭式系统或系统架构中引入该技术以保障系统的安全也是一种必然的趋势.

7　结　论

本文立足于 TrustZone 技术特性、学术研究及其应用研究 3 个方面，对该技术作了系统的概述. 首先介绍了该技术的硬件和软件架构，并且对其安全扩展作了详细分析. 接着对其安全机制作了深入的剖析，指出其如何基于硬件和软件架构实现系统范围的安全. 基于此基础上将该技术的优势与不足进行了分析，并针对不足给出了解决方案. 文中也将该技术与其他提高嵌入式系统安全性技术进行了分析对比，同时对如何基于该技术的系统安全架构构建符合 GP 标准的可信执行环境作了整体概述，接下来从应用的角度出发，总结概括了基于该技术的现有国内外研究和商业应用情况. 最后，本文结合业界现有的基于该技术的研究成果，展望该技术应用发展方向.

致　谢　评审专家和编辑部老师为本文提出了宝贵意见和建议，作者在此表示衷心的感谢!

参　考　文　献

[1] Ravi S, Raghunathan A, et al. Security in embedded systems: Design challenges. ACM Transactions on Embedded Computing Systems, 2004, 3(3): 461-491

[2] Kumar K. DRM on Android//Proceedings of the 10th IEEE India Conference. Mumbai, India, 2013: 1-6

[3] Alves T, Felton D. TrustZone: Integrated hardware and software security enabling trusted computing in embedded system. Government Information Quarterly, 2004, 3(4): 18-24

[4] Anwar W, Lindskog D, et al. Redesigning secure element access control for NFC enabled Android smartphones using mobile trusted computing//Proceedings of the 2013 International Conference on Information Society. Toronto, Canada, 2013: 27-34

[5] ARM Limited. ARM Security Technology Building a Secure System using TrustZone Technology. White Paper, 2009

[6] Lee R B, Kwan P C S, et al. Architecture for protecting critical secrets in microprocessors//Proceedings of the 32nd International Symposium on Computer Architecture. Madison, USA, 2005: 2-13

[7] Wilson P, Frey A, et al. Implementing embedded security on dual-virtual-CPU systems//Proceedings of the Design and Test of ICs for Secure Embedded Computing. Shilong, India, 2007: 582-591

[8] An Yang, Zhao Bo, Li Hong-Bo. Extension implementation of TCM in the embedded system based on FPGA//Proceedings of the 2013 International Conference on Computer Science and Application. Wuhan, China, 2013: 749-752

[9] Lie D, Thekkath C, et al. Architecture support for copy and tamper-resistant software//Proceedings of the ASPLOS-IX 2000. Massachusetts, USA, 2000: 1-10

[10] Suh G E, Clarke D, et al. AEGIS: Architecture for tamper-evident and tamper-resistant processing//Proceedings of the 27th International Conference on Supercomputing. San Francisco, USA, 2003: 1-12

[11] Li Hong-Juan, Lan Yu-Qing. A design of trusted operating system based on Linux//Proceedings of the 2010 International Conference on Electrical and Control Engineering. Wuhan, China, 2010: 4598-4601

[12] Ukil A, Sen J, et al. Embedded security for Internet of Things//Proceedings of the 2nd National Conference on Emerging Trends and Applications in Computer Science. Shilong, India, 2011: 1-6

[13] ARM Limited. ARM TrustZone API Specification Version 3.0. White Paper, 2009

[14] Gilmont T, Legat J-D, Quisquater J-J. Enhancing security in the memory management unit//Proceedings of the 25th EUROMICRO Conference. Milan, Italy, 1999: 449-456

[15] GlobalPlatform Inc. GlobalPlatform Device Technology TEE System Architecture Version 1.0. White Paper, 2012

[16] GlobalPlatform Inc. GlobalPlatform Device Technology TEE Client API Specification Version 1.0. White Paper, 2010

[17] GlobalPlatform Inc. GlobalPlatform Device Technology TEE Internal API Specification Version 1.0. White Paper, 2011

[18] GlobalPlatform Inc. GlobalPlatform Device Technology Trusted User Interface API Version 1.0. White Paper, 2013

[19] Wang Xi-You. The Research and Application of ARM TrustZone Security Isolation Technology [M. S. dissertation]. University of Electronic Science and Technology of China, Chengdu, 2010(in Chinese)
(王熙友. ARM TrustZone 安全隔离技术研究与应用[硕士学位论文]. 电子科技大学, 成都, 2010)

[20] Li Wen-Hao, Ma Ming-Yang, et al. Building trusted path on untrusted device drivers for mobile devices//Proceedings of the 5th Asia-Pacific Workshop on Systems. Beijing, China, 2014: 1-7

[21] Li Wen-Hao. T6, an operating system for TrustZone based trusted execution environment (TEE) in ARM-based systems. White Paper, 2014

[22] Fitzek A. Development of an ARM TrustZone Aware Operating System ANDIX OS [M. S. dissertation]. Graz University of Technology of Austria, Styria, Austria, 2014

[23] Giesecke & Devrien Inc. G&D announces MobiCore integrated security platform to support Samsung GALAXY S III in Europe. White Paper, 2012

[24] Sierraware. Open Virtualization Build and Boot Guide for ARM V7 and ARM V8. White Paper, 2014

[25] Texas Instruments. M-Shield Mobile Security Technology: Making Wireless Secure. White Paper, 2008

[26] Intel Corporation. Intel Trusted Execution Technology: Hardware-based Technology for Enhancing Server Platform Security. White Paper, 2012

[27] AMD. Inc. AMD64 Virtualization Codenamed "Pacifica" Technology: Secure Virtualization Machine Architecture Reference Manual. White Paper, 2005

[28] McKeen F, Alexandrovich I, et al. Innovative instruction and software model for isolated execution//Proceedings of the 2nd International Workshop on Hardware and Architecture Support for Security and Privacy. New York, USA, 2013: 1-8

[29] Anand V, Saniie J, et al. Threat-adaptive architecture for trusted platform modules in secure computing systems//Proceedings of the 2010 IEEE International Conference on Electro/Information Technology. lllinois, America, 2010: 1-6

[30] Pirker M, Slamanig D. A framework for privacy-preserving mobile payment on security enhanced ARM TrustZone platforms//Proceedings of the 11th International Conference on Trust, Security and Privacy in Computing and Communications. Merseyside, UK, 2012: 1155-1160

[31] Großschädl J, Vejda T, et al. Reassessing the TCG specification for trusted computing in mobile and embedded systems//Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust. Anaheim, USA, 2008: 84-90

[32] Smalley S, Craig R. Security enhanced (SE) Android: Bring flexible MAC to Android//Proceedings of the 20th Annual Network and Distributed System Symposium. San Diego, USA, 2013: 1-18

[33] Zheng Chao-Wen. Overview of security enhanced Android's security architecture//Proceedings of the 2nd Conference on Teaching and Computational Science. Perugia, Italian, 2014: 48-50

[34] SamSung Inc. An Overview of SamSung KNOX, White Paper, 2014

[35] AMD Inc. AMD Safe Technology. White Paper, 2010

[36] Apple Inc. White Paper iOS Security. White Paper, 2014

[37] Azab A M, Ning Peng, et al. Hypervision across worlds: Real-time kernel protection from the ARM TrustZone secure world//Proceedings of the 21st ACM Conference on Computer and Communication Security. New York, USA, 2014: 90-102

[38] Ou Qing-Yu, Luo Fang, et al. High-security system primitive for embedded systems//Proceedings of the International Conference on Multimedia Information Networking and Security. Hubei, China, 2009: 319-321

[39] Xu Yan-Ling, Pan Wei, Zhang Xin-Guo. Design and implementation of secure embedded system based on TrustZone//Proceedings of the 2008 International of Secure Embedded Software and Systems. Sichuan, China, 2008: 136-141

[40] Sun He, Sun Kun, et al. TrustDump: Reliable memory acquisition on smartphones//Proceedings of the 19th European Symposium on Research in Computer Security. Wroclaw, Poland, 2014: 202-218

[41] Winter J. Trusted computing building blocks for embedded Linux-based ARM TrustZone platforms//Proceedings of the 3rd ACM Workshop on Scalable Trusted Computing. New York, USA, 2008: 21-30

[42] Dietrich K, Winter J. Implementation aspects of mobile and embedded trusted computing//Proceedings of the 2nd International Conference on Trust & Trustworthy Computing. Oxford, UK, 2009: 29-44

[43] Feng Wei, Feng Deng-Guo, et al. TEEM: A user-oriented trusted mobile device for multi-platform security applications //Proceedings of the 6th International Conference on Trust & Trustworthy Computing. London, UK, 2013: 133-141

[44] Hussin W H, Coulton P, Edwards R. Mobile Ticketing system employing TrustZone technology//Proceedings of the International Conference on Mobile Business. Sydney, Australia, 2005: 651-654

[45] Shin J, Kim Y, et al. DFClound: A TPM-based secure data access control method of cloud storage in mobile devices//Proceedings of the 4th International Conference on Cloud Computing Technology and Science. Taipei, China, 2012: 551-556

[46] Hussin W H W, Edwards R, et al. E-pass using DRM in Symbian v8 OS and TrustZone: Securing vital data on mobile devices//Proceedings of the 2006 International Conference on Mobile Business. Copenhagen, Denmark, 2006: 1-5

[47] Mihm T. Protecting critical data//Proceedings of the Design and Test of ICs for Secure Embedded Computing. New York, USA, 2007: 592

[48] Zhao Shi-Jun, Zhang Qian-Ying, et al. Providing root of trust for ARM TrustZone using on-chip SRAM//Proceedings of the 4th International Workshop on Trustworthy Embedded Devices. New York, USA, 2014: 25-36

[49] Guajardo J, Kumar S S, et al. FPGA intrinsic PUFs and their use for IP protection//Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems. Vienna, Austria, 2007: 63-80

[50] Adnan L H, Yussoff Y M, et al. Secure boot process for wireless sensor node//Proceedings of the International Conference on Computer Application and Industrial Electronics. Kuala Lumpur, Malaysia, 2010: 646-649

[51] González J, Hölzl M, et al. A practical hardware-assisted approach to customize trusted boot for mobile devices//Proceedings of the 17th International Conference on Information Security. Hong Kong, China, 2014: 542-554

[52] Azab A M, Ning Peng, et al. HIMA: A hypervisor-based integrity measurement agent//Proceedings of the 25th Annual Computer Security Applications Conference. Oahu, Honolulu, 2009: 461-470

[53] Criswell J, Lenharth A, et al. Secure virtual architecture: A safe execution environment for commodity operating systems //Proceedings of the 21st ACM SIGOPS Symposium on Operating Systems Principles. New York, USA, 2001: 351-366

[54] Garfinkel T, Pfaff B, et al. Terra: A virtual machine-based platform for trusted computing//Proceedings of the 19th ACM Symposium on Operating Systems principle. New York, USA, 2003: 193-206

[55] Oh Soo-Cheol, Koh KwangWon, et al. Accelerating of dual OS virtualization in embedded systems//Proceedings of the 7th International Conference on Computing and Convergence Technology. Seoul, South Korea, 2012: 1098-1101

[56] Cereia M, Bertolotti I C. Asymmetric virtualization for real-time systems//Proceedings of the IEEE International Symposium on Industrial Electronics. Cambridge, UK, 2008: 1680-1685

[57] Sangorrin D, Honda S, et al. Dual operating system architecture for real-time embedded systems//Proceedings of the 6th International Workshop on Operating Systems Platforms for Embedded Real-Time Applications. Brussels, Belgium, 2010: 6-15

[58] Zhang Yu-Qing, Wang Kai, Yang Huan, et al. Android security review. Journal of Computer Research and Development, 2014, 51(7): 1385-1396(in Chinese)
(张玉清，王凯，杨欢等. Android 安全综述. 计算机研究与发展，2014，51(7)：1385-1396)

[59] Vargas R J G, Huerta R G, et al. Security controls for Android//Proceedings of the 4th International Conference on Computational Aspects of Social Networks. Sao Carlos, Portugal, 2012: 212-216

[60] Kim S W, Lee C, et al. Secure device access for automotive software//Proceedings of the 2013 International Conference on Connected Vehicles and Expo. Las Vegas, USA, 2013: 177-181

[61] Winter J. Experimenting with ARM TrustZone or: How I met friendly piece of trusted hardware//Proceedings of the 11th International Conference on Trust, Security and Privacy in Computing and Communications. Liverpool, UK, 2012: 25-27

[62] Winter J, Wiegele P, et al. A flexible software development and emulation framework for ARM TrustZone//Proceedings of the 3rd International Conference on Trusted Systems. Beijing, China, 2011: 1-15

ZHENG Xian-Yi, born in 1986, Ph. D. candidate. His main research interests include computer architecture, network and system security.

LI Wen, born in 1976, Ph. D., associate professor. His main research interests include computer architecture and embedded system.

MENG Dan, born in 1965, Ph. D., professor, Ph. D. supervisor. His main research interests include computer architecture, cloud computing, network and system security.

Background

Embedded system security has become a focus attention in the field of information security. ARM as a mainstream IP supplier of embedded processors has proposed TrustZone technology to achieve a set of system level security solutions. It has caused wide public concern because it improves system security but not affecting the original processor design.

So far, researchers around the world have done some research work on this topic and have proposed some security architectures and strategies based on TrustZone Technology. It is a pity that none of them have yet made any efforts on analyzing and evaluating the technology. Therefore, it is a common phenomenon that people including enterprises and individuals often construct some security platforms and develop security services based on TrustZone. However, it is more important how to apply the technology to more system architectures and more application fields, but it also makes more difficulties to researchers and developers.

In this paper, we firstly introduce hardware and software architecture of TrustZone technology and analyze its security extensions in detail. Then, we make thoroughly analysis to its security mechanism and point out how to implement system level security based on hardware and software. We also analyze and compare it to other technologies of improving embedded system security. At the same time, we have an overall overview how to build TEE based GP standards, and we make a summary to the existing researches and commercial applications at home and abroad from application perspective. Finally, we prospect the development direction of the technology combining with the existing research results.

This work is supported by the National High Technology Research and Development Program (863 Program) of China under Grant No. 2012AA01A401 and the National Science and Technology Major Project of China under Grant No. 2013ZX01029003-001.