物联网操作系统安全研究综述

彭安妮[1]，周威[1]，贾岩[2]，张玉清[1,2]

（1. 中国科学院大学国家计算机网络入侵防范中心，北京 101408；
2. 西安电子科技大学网络与信息安全学院，陕西 西安 710071）

摘 要：随着物联网的迅速普及和应用，物联网系统核心（操作系统）的安全问题越发显得急迫和突出。首先，对现阶段市场上广泛应用的物联网操作系统及其特征进行了介绍，分析了其与传统嵌入式操作系统的异同；然后，在调研和分析大量物联网操作系统相关文献的基础上，从构建完整安全系统的角度对现有物联网操作系统安全研究成果进行有效的分类和分析；进一步指出了物联网操作系统安全所面临的挑战和机遇，总结了物联网操作系统安全的研究现状；最后，结合现有研究的不足指出了物联网操作系统安全未来的热点研究方向，并特别指出了物联网系统生存技术这一新的研究方向。

关键词：物联网；安全；操作系统

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000-436x.2018040

Survey of the Internet of things operating system security

PENG Anni[1], ZHOU Wei[1], JIA Yan[2], ZHANG Yuqing[1,2]

1. National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing 101408, China
2. School of Cyber Engineering, Xidian University, Xi'an 710071, China

Abstract: With the rapid popularization and wide application of the Internet of things (IoT), the security problems of IoT operating system, which is the essential part, become more and more urgent. Firstly, the famous IoT operating systems and their different features were introduced, then it was compared with present embedded systems. Secondly, On the basis of the survey of research related to IoT operating system, the research was discussed and analyzed from the view of building a comprehensive security system, then security challenges and opportunities which the IoT system faced were pointed out, and the research status of the security of the IoT operating system was summarized. Finally, the promising future study directions in the IoT operating system security field were discussed based on the drawbacks of the existing researches, particularly, the IoT system survival technology as a new research direction was pointed out.

Key words: Internet of things, security, operating system

1 引言

数据表明，物联网逐渐成为继计算机、互联网之后，世界信息产业发展的第 3 次浪潮，并得到各个国家与企业的高度重视，发展十分迅速。根据 Statista 门户网站的最新统计数据[1]，物联网市场规模不断扩大，设备数目高速增长，互联设备数量 2016 年已经达到 176 亿，预计到 2020 年将突破 300 亿。

收稿日期：2017-10-12；**修回日期**：2018-03-06

基金项目：国家重点研发计划基金资助项目（No. 2016YFB0800703）；国家自然科学基金资助项目（No. 61572460, No. 61272481）；信息安全国家重点实验室的开放课题基金资助项目（No.2017-ZD-01）；国家发改委信息安全专项基金资助项目（No. (2012)1424）；国家“111”计划基金资助项目（No.B16037）

Foundation Items: The National Key Research and Development Program of China (No.2016YFB0800703), The National Natural Science Foundation of China (No.61572460, No.61272481), The Open Project Program of the State Key Laboratory of Information Security(No.2017-ZD-01), The National Information Security Special Projects of National Development and Reform Commission of China (No.(2012)1424), 111 Project Foundation of China (No.B16037)

而物联网操作系统作为物联网行业发展的关键技术，其发展趋势也十分迅猛。目前，ARM、谷歌、微软、华为、阿里等国内外公司均推出了物联网操作系统[2]。

由于物联网存在设备的异构性、设备间的互用性以及部署环境的复杂性等因素，物联网应用普遍安全性较低、不便于移植、成本较高。其中，物联网操作系统作为连接物联网应用与物理设备的中间层，对解决这些问题起着主要作用。物联网操作系统可以屏蔽物联网的碎片化特征，为应用程序提供统一的编程接口，从而降低开发时间和成本，便于实现整个物联网统一管理。鉴于物联网操作系统作为物联网系统架构的核心，其安全问题将会严重影响整个物联网生态系统，所以物联网操作系统也逐渐成为攻击者的重点目标。

近年，随着物联网应用领域扩大，物联网系统安全问题愈发严重。例如，2010 年曝光的“震网病毒”，攻击者利用其入侵多国核电站、水坝、国家电网等工业与公共基础设施的操作系统，造成了大规模的破坏[3]。2016 年，爆发的现今最大规模的“IoT 僵尸网络 Mirai”，其控制物联网设备的方法除了利用默认的用户名口令，还主要利用了物联网设备中的系统漏洞如缓冲区溢出等，从而控制了大量的物联网设备。

随着物联网设备与应用逐渐增多，物联网操作系统面临的安全风险也逐渐增大。任何一个存在系统漏洞的物联网设备，都会给整个物联网系统带来潜在的安全威胁，因此，亟待提出能更加有效保护物联网操作系统的安全机制。然而现阶段关于物联网操作系统安全研究的文献较少，已有的研究成果也存在严重的不足。物联网设备、通信协议和应用场景的多样化与异构性也使对物联网操作系统很难构建一个系统的安全体系。为了使研究人员更加清楚地了解物联网安全研究现状，促进物联网操作系统安全发展，本文对物联网操作系统安全现状进行了深入分析，指出了挑战和机遇以及未来的研究方向，主要贡献如下。

1) 对现有典型物联网操作系统进行了全面调研与分析，总结了物联网操作系统的关键新特性与存在的安全问题，并根据不同的物联网应用场景提出了相应的安全需求。

2) 通过调研大量的现阶段物联网操作系统相关研究成果，将物联网操作系统安全相关文献按照“系统安全构建—系统安全性分析—系统攻击防御”这 3 个角度进行了分类，并将不同安全技术对应到不同的应用场景需求，从而可以更加清晰、全面地了解物联网操作系统安全研究现状。

3) 结合现有物联网操作系统的安全问题与研究现状，深入分析导致安全问题产生的根本原因，并指出了物联网操作系统安全研究中面临的挑战与机遇。

4) 结合物联网操作系统安全发展中的挑战与机遇，为相关研究者指出未来的热点研究方向，特别指出了物联网系统生存技术这一新的研究方向。

2　背景介绍

2.1　物联网设备系统架构

现阶段物联网设备与应用虽然多种多样，但其操作系统主要由各种嵌入式操作系统改进而来，所以其逻辑架构层次本质与嵌入式系统架构是相似的，如图 1 所示。

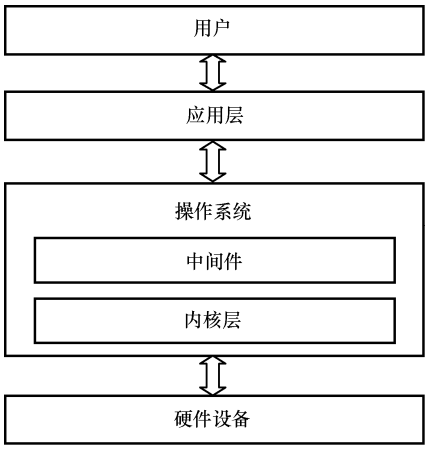

图 1　物联网系统架构

需要注意的是，物联网与嵌入式系统架构具有 2 个不同：1) 物联网系统架构中各个层次并不是固定的，如工业和医疗领域的某些控制设备，其自身受资源限制可能并没有操作系统层，只是通过远程应用的命令直接进行控制，另外，还有许多轻量级嵌入式物联网设备，其应用直接与简化的 RTOS 进行交互并没有中间件层；2) 物联网硬件设备、操作系统和应用经常在物理上也是可分离的，如智能家居中的传感器、IP 摄像头等，其物理设备上只具备简单的操作系统，而其应用则是在远程移动设备或者云服务器上。本文主要讨论系统架构中操作系统层次的安全问题与安全研究现状。

2.2 物联网操作系统特征与安全问题

为了更好地保证物联网操作系统安全，保障物联网设备工作的正常高效，首先应该了解物联网操作系统新的特征。其特征使物联网操作系统能够与物联网的其他层次结合得更加紧密，数据共享更加方便，同时也是影响物联网操作系统安全的主要因素。本文在调研现有物联网操作系统后，选取有代表性的 10 个物联网操作系统，总结其各自主要特性，如表 1 所示。表 1 进一步提炼出物联网操作系统 5 个重要特征，表 2 对物联网操作系统和嵌入式操作系统的主要特征进行了比较。

表 1 物联网 10 个操作系统及其特性

操作系统	特性
文献[4]（RIOT）	支持平台较多，能在多平台（如嵌入式设备和传感器等）上运行，较容易开发
文献[5]（Contiki）	是一个开源的、容易移植的多任务操作系统，适用于内存资源受限的设备
文献[6]（Android Things）	使用 Weave 的通信协议，实现设备与云端相连，并且与谷歌助手等服务交互
文献[7]（ARM mbed）	ARM 处理器专用，采用事件驱动的单线程架构，可用于尺寸小、低功耗的物联网设备
文献[8]（Nucleus RTOS）	兼容性强，为众多嵌入式架构提供了有力的支持
文献[9]（FreeRTOS）	轻量级物联网实时操作系统
文献[10]（Lite OS）	华为公司开发的轻量级的物联网操作系统，具备零配置、自组网、跨平台的能力
文献[11]（Green Hills Integrity）	具备高性能、安全性和可靠性
文献[12]（eLinux）	嵌入式 Linux 操作系统，该操作系统基于 Linux 内核，支持该操作系统的厂商、芯片和产品比较广泛
文献[13]（Tizen）	具有很强的移植性，可用于手机、电脑、智能电视、车载系统等多种智能设备

1) 硬件驱动和操作系统内核可分离性。由于物联网设备异构性较大，不同的设备会有不同的固件与驱动程序，所以对操作系统内核与驱动的可分离性要求更高，进而提高操作系统内核的适用性和可移植性。

2) 可配置剪裁性。物联网终端的硬件配置各种各样，有小到十几千字节内存的微型嵌入式应用，也有高达几十兆字节内存的复杂应用领域。因此，对物联网操作系统可裁剪性和配置性的要求比对传统嵌入式操作系统要求更高，同一个操作系统，通过裁剪或动态配置，既能够适应低端的需求，又能够满足高端复杂的需求。

3) 协同互用性。传统的嵌入式系统大多独立完成某个单一的任务，而在物联网环境下各种设备之间相互协同工作的任务会越来越多，所以对物联网操作系统之间通信协调的要求会越来越高。

4) 自动与智能化。随着物联网应用技术的发展，物联网设备需要人为干预的操作越来越少，而自动化与智能化的操作越来越多，所以物联网操作系统比传统的嵌入式操作系统更加智能。

5) 安全可信性。传统工业设备的嵌入式操作系统单独处于封闭环境中，同时传统的嵌入式设备与用户的关联并不那么紧密。而随着物联网设备在工业与生活中的普遍应用，其将会面临更加严重的网络攻击威胁，同时物联网设备存储和使用的数据更加敏感和重要。这些系统被控制后将对个人、社会和国家安全造成严重威胁，因此，对于物联网设备的安全和可信性要求越来越高。

表 2 物联网操作系统与传统嵌入式系统特征比较

特征	物联网操作系统	传统嵌入式系统
专用性	较高	高
可配置剪裁性	高	较高
协同互用性	高	较低
硬件驱动与操作系统内核可分离性	高	较低
自动与智能化	高	较低
安全可信性	高	较高

本文进一步分析了上述物联网操作系统的安全设计，指出了其主要存在的 3 个问题。

1) 直接沿用原有的安全机制。例如，Android Things 直接沿用了 Android 系统的一些基础安全机制，并没有深入分析物联网设备实际的软硬件特性与需求，还有 eLinux 也主要是基于 Linux 内核安全机制，并没有为物联网设备设计额外的安全机制。

2) 缺乏对终端系统安全设计。现有的物联网操作设计时普遍只关注其功能要求。例如，Contiki 主要为实时性做了优化设计，RIOT 主要为支持各种通信协议进行了改进。大多并没有考虑对系统安全进行额外的设计。即使像 mbed 操作系统，虽然在设计时考虑安全因素，但其主要安全保护措施是为了保护通信安全如 SSL，但对系统本身还没有采取有效的防护措施。

3) 没有充分利用设备自身硬件架构安全特性。上述物联网操作系统如 FreeRTOS、RIOT、Tizen

等普遍是运行在ARM Cortex-M系列的CPU核心上的。但是对于 Cortex-M 自身提供的硬件安全机制如内存保护单元（MPU），在这些操作系统设计中却没有具体的应用。而这些自带的硬件安全机制如果进行合理的配置和使用，可在不增加额外硬件配置的条件下实现高效的系统防御措施。

2.3　不同物联网场景下操作系统安全需求

现阶段物联网应用场景逐渐增多，不同场景下的需求不同，设备软硬件资源存在差异，故各应用场景对应的系统安全需求侧重点也不相同。在介绍物联网操作系统安全研究现状之前，本节首先对各个应用场景的安全需求进行分析简述，只有明确其安全需求，才能采取有针对性的安全机制。

2.3.1　智能家居

在智能家居越发普及的同时，各种智能家居设备系统中保存和使用的用户隐私信息也越来越多。这些数据不仅包含与用户身份认证直接相关的指纹、密码等隐私信息，还包括用户日常生活中的隐私信息，例如，温度传感器记录了家中各个房间的实时温度信息；智能电表记录了家中的用电情况等。而且目前用户隐私保护意识较差，智能家居产品也缺乏隐私数据使用规范，导致智能家居设备隐私数据泄露日趋严重[14]。

智能家居操作系统的首要安全需求是保护用户的隐私数据，操作系统需要在不影响应用端使用这些隐私数据的同时防止隐私数据泄露。

2.3.2　智能医疗

在智能医疗场景下，设备收集的用户隐私信息会更多，同时智能医疗设备的隐私信息会共享给诸多医疗单位，加剧用户医疗隐私信息泄露的风险[15]。另一方面，该场景下设备运行的稳定性需要得到保证，医疗设备尤其是胰岛素泵[16]、心脏起搏器[17]等人体嵌入式设备尤为重要，一旦这些医疗设备的操作被恶意控制，将会直接威胁用户的生命安全，针对智能医疗设备[18]的勒索软件也开始逐渐增多。

对于智能医疗设备的操作系统，一方面需要对收集、使用和传输的隐私数据进行严格保护，另一方面需要对设备的关键程序操作也进行实时的监控，在异常行为最终执行之前采取对应的处理措施，切实保障智能医疗设备的安全运行。

2.3.3　智能工业

现阶段工业生产中应用的物联网设备越来越多，这些物联网设备在方便企业进行更加智能自动化管理和操作的同时，也扩大了其攻击面。例如，“震网病毒”等对关键工业设施的攻击会对企业和国家产生严重危害。

因此，关键智能工业设备操作系统最重要的安全需求应该是对其控制程序的完整性和可信性的验证。确保控制可信命令得到执行，同时，对于设备的异常行为做到提早发现和快速处理，防止异常程序行为的执行。另外，对关键工业设备的外围接口也要进行安全隔离，防止通过如 U 盘等外围设备插入关键控制设备传播恶意代码。

2.3.4　智能汽车

随着市场上联网的智能汽车逐渐增多，现实中对智能汽车的电子攻击也层出不穷[19]。智能汽车的系统漏洞也逐渐成为不法者盗取汽车的重要手段[20]。另一方面，用户个人车辆行驶数据具有较大的商业价值，也成为不法者和各大公司窃取的主要目标。

对于智能汽车操作系统，一方面要防止其存储的车辆行驶隐私数据在用户不知情的情况下泄露，另一方面要对车辆系统的控制总线 CAN-Bus 进行特别防护和隔离，防止攻击者借助安全性较低的系统程序（如车载娱乐系统、导航系统等）对其非法访问。另外，智能汽车的安全防护措施必须满足车辆在实际使用时实时性的要求，对关键行驶控制设备必须进行实时监控，及时终止异常行为执行。

3　物联网操作系统安全研究现状

本文首先广泛调研了物联网安全的相关文献，发现其中涉及物联网操作系统安全的研究文献较少且研究技术较为分散。于是进一步调研了移动操作系统安全领域中可用于物联网环境下的具体安全技术（如可信执行环境隔离、安全启动等），最终整理出多篇物联网操作系统安全相关研究文献，并对其进行了更加深入的分析与研究。

从这些研究文献中发现，物联网操作系统安全还处于初级阶段，对相关研究的安全技术也没有恰当的分类体系。所以本文从安全系统构建的过程出发，依据“系统安全构建—系统安全性分析—系统攻击防御”这一过程对现有物联网操作系统主流安全技术进行分类，如图 2 所示。首先，在操作系统构建之初就应尽可能全面地考虑到其安全问题，分析需求并设计相应方案，这比构建一个脆弱的系统再进行漏洞修补的方案更为高效，能起到事半功倍

的效果。然而，由于操作系统的开放性，构建一个永久安全的系统也是无法实现的。所以在系统构建后，及时地对系统进行安全分析发现安全问题也显得十分重要。同时，由于攻击者手段和能力不断提高，原有设计难以应付日新月异的攻击手段，物联网操作系统侦测攻击的能力也需要不断提升，来抵御各种潜在的系统攻击。

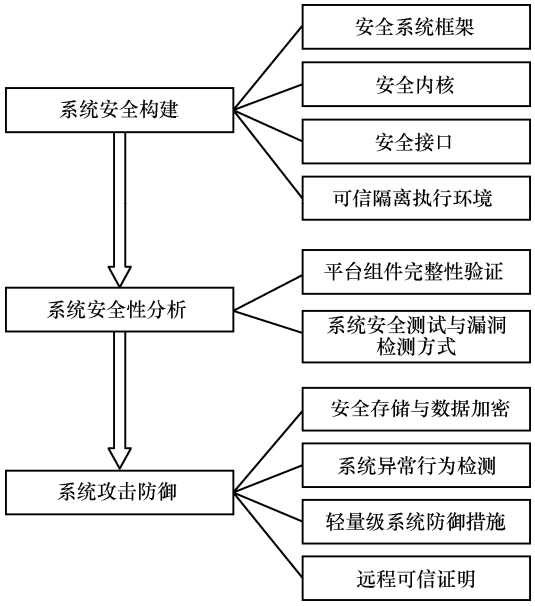

图 2 物联网操作系统中的安全研究分类

本文将分别按系统安全构建、系统安全性分析、系统攻击防御这3个方面对现有物联网操作系统研究工作进行讨论。使相关安全研究人员可以更加直观地了解物联网操作系统构建与使用过程中已有的安全问题和技术，便于快速开展进一步的研究。另外，由于现有关于安全接口设计的研究内容很少，所以本文对安全接口相关研究内容没有展开讨论。

3.1 物联网操作系统安全构建

实现物联网操作系统安全的首要步骤是在系统设计之初就尽可能全面地考虑到其安全问题，构建一个相对稳固的、安全的系统。对于实现物联网操作系统安全构建的研究，本节从系统安全框架构建、安全内核设计和可信隔离执行环境构建这 3 个方面展开。

3.1.1 安全系统框架

目前，市场上的物联网产品在设计阶段往往忽视安全因素，导致现阶段物联网产品普遍存在大量的安全漏洞。而如果在设计之初就考虑安全因素，采用更为安全的操作系统设计框架，会有效缓解这一问题。因此，学术界很早就开始关注操作系统安全框架的设计，目前主要提出 2 个指导性意见，同样适用于物联网操作系统。

1) 支持用户自定义控制系统，在设计时应该让用户拥有自主选择信任范围的权利，而不能盲目相信设备厂商提供的系统或固件。

2) 对系统提出安全防御措施时要尽可能地减少安全测试复杂度。现阶段一些数据加密、安全启动等安全措施在抵御攻击者的同时，也给设备的安全测试和分析增加了难度[21]。

根据上述指导意见，有研究人员提出在物联网设备中内置安全模块，为用户提供动态检测、诊断、隔离等安全功能，从而使用户摆脱对厂商的依赖，拥有检测设备安全和可信的能力[22]。还有研究人员根据现阶段物联网产品存在的安全问题，有针对性地提出了安全产品设计建议[23]，值得学习借鉴和参考。

3.1.2 安全内核

内核是操作系统的核心部分，用于完成如进程调度、内存管理等主要功能。对于物联网设备而言，其操作系统本身就十分简洁，绝大多数功能均通过内核来实现。因此，设计安全内核对于物联网操作系统安全构建显得十分重要。

目前，许多研究致力于轻量级安全内核的研究，其工作主要可分为 2 个方面。一方面是直接改进原有内核的设计增加安全性。例如，有研究人员设计了安全内核原型系统[24]，其可以提供安全认证、访问控制以及授权管理等多种安全功能并可适用于多种嵌入式操作系统；还有研究人员将原有整体内核进行分区隔离，来有效防止攻击的传播和扩散，但安全分区间的通信会额外增加系统负担[25]。改进内核的设计方法虽然可以直接提高内核的安全性，但这无疑增加了开发人员的负担，通用内核难以直接应用。

另一方面，研究人员致力于通过增加额外的模块来对原有内核进行监测和验证。例如，有研究人员设计了独立的、轻量级的可信执行环境，用于保护原有内核的关键操作[26]；还有研究人员增加了额外的验证模块用于实时动态地验证原有内核的安全性，从而保证内核关键操作与通信的正确运行[27]。

另外，实时性也是物联网操作系统必备的特性之一，但其与安全性很难兼备，所以 Malenko 等[28]

提出了适用于物联网设备的实时操作系统内核设计安全要求，如完整性、机密性、可用性、可靠性、可维护性等，为后续物联网实时操作系统设计提供了很好的参考。

3.1.3　可信隔离执行环境

随着物联网设备在工业与关键基础设施中的应用愈发广泛，其安全威胁也逐步增加。但如果对系统所有层攻击都对应采取相应的防御措施会使开销过大，且防御措施也难以面面俱到。所以在系统构建时，设计可信隔离执行环境用于关键程序执行。

可信执行环境的构建主要通过硬件和软件这 2 种方式来实现。对于硬件隔离，有研究人员提出利用安全协处理器来确保工作的独立执行[29]。然而，使用安全协处理器会有额外功耗以及具有较低的计算性能等缺点，因此，Petroni 等[30]提出只将协处理器用于侦测和保证中央处理器实际计算的完整性。但由于侦测的工作方式是周期性的，所以其工作间隙的攻击无法被侦测到；同时轻量级物联网设备一般并不具备协处理器或额外的硬件配置。如何利用最少的设备和已有的硬件资源实现可信执行环境的构建仍需进一步研究。

对于软件隔离，研究人员主要通过软件错误隔离（SFI, software fault isolation）和硬件虚拟化来实现。软件错误隔离主要是在原有程序中增加对控制流完整性的检查，并对使用的内存进行访问控制[31]，从而实现应用之间控制流与数据流的相互隔离。还有研究人员通过增加虚拟化层来实现操作系统与关键应用程序之间的相互隔离，关键应用程序的内存数据直接由虚拟层进行管理，而操作系统无法对其修改和查看。故即使在虚拟层运行的客户系统被攻击者控制，其仍然无法影响系统可信代码的执行[32,33]。但软件操作隔离的方法需要增加额外的检测程序，会降低原有轻量级物联网操作系统的工作效率；而虚拟化的方法一般需要处理器硬件架构的支持，物联网设备大多也不具备这样的配置。所以更加有效的轻量级可信执行环境构建方法还有待更加深入的研究。

3.2　物联网操作系统安全性分析

一个永久安全的系统是不存在的，尤其对于关键工业与基础设施的物联网设备需要严格保证其数据与操作的可信性，这就需要对物联网操作系统进行安全性分析，验证设备安全的同时可以及时发现与修复系统安全问题。关于物联网操作系统的安全性分析的研究，本节将从平台组件完整性验证、系统安全测试与漏洞检测方法这 2 个方面来阐述。

3.2.1　平台组件完整性验证

由于物联网设备的多样性，各种物联网设备厂商都会对其设备定制平台组件，导致现阶段系统组件碎片化严重。如何确保各种平台组件的安全性成为物联网操作系统安全性分析的一大难点。现有研究主要通过平台组件完整性验证来及时发现被恶意修改的平台组件，从而保护系统安全。对于平台组件完整性的验证主要可分为安全启动、运行时验证和更新验证 3 个部分。

安全启动主要通过验证各启动模块的数字签名（主要由模块代码散列值和设备厂商提供的私钥组成）并结合可信计算基（TCB, trusted computing base）来保证不可修改的启动顺序。具体验证过程首先由硬件 TCB 将系统最先启动的模块（如 BootLoader）加载到内存进行验证，验证通过再加载下一模块（如内核）对其进行验证，以此类推，其中任何一个启动模块验证失败都会导致安全启动终止，只有所有模块均按顺序通过验证后才可以完成安全启动[34]。安全启动的相关技术现阶段越发成熟并已经广泛应用于大量的移动设备中，例如，ARM 公司推出的 TrustZone 架构就携带了安全启动的功能。

要确保平台组件的完整性，只通过在启动阶段验证是远远不够的，攻击者还可在系统启动后对平台组件进行恶意修改，所以需要在系统运行阶段对平台组件完整性进行验证。现阶段主要是通过一个额外的监测程序不断地对平台组件代码进行验证[30]，并尝试自动修复被恶意篡改的平台组件[35]。该监测程序自身完整性可通过设备密钥对其数字签名进行验证，但这无疑会增加系统运行时额外的开销。对系统资源十分有限的物联网设备的运行，平台组件完整性验证方法还有待改进。

另外，平台组件由于功能增加或安全漏洞修复会经常需要更新，故需要验证更新组件的完整性与可信性，从而防止攻击者通过假冒更新组件安装恶意程序。Kohnäuser 等[36]提出利用无线网络中的其他设备来验证微型嵌入式系统平台更新代码可信性方案，即在网络中，各设备远程平台代码在更新后，进行互相验证，从而排除处于不可信状态的设

备，大大提高了攻击者伪造平台组件更新的难度。但现阶段对单一物联网设备进行安全平台组件更新的方法研究较为稀少。

3.2.2 系统安全测试与漏洞检测方法

目前，安全问题在物联网设备系统中十分普遍，Costin 等[37]在静态分析了大量物联网设备系统固件及其更新补丁的源码后，发现了许多已知和未知的安全漏洞，例如，未保护的后门私钥泄露问题、存在于通过 Wi-Fi 连接的 Web 服务中的 XSS 漏洞等。因此，对设备本身进行安全测试与漏洞挖掘是十分必要的。

由于物联网设备的异构性，其安全测试与漏洞挖掘方法很难统一，虽然 2016 年 Sachidananda 等[38]第一个提出了可以应用于不同种类物联网设备的测试框架，但其主要针对已知的设备系统漏洞，并且缺乏对实际产品的大量测试。同年，Mer 等[39]提出在智能医疗场景中从设备系统端到云端的完整测试框架，但其主要方法依靠静态分析缺乏动态测试并且测试方法过于简单。Tabrizi 等[40]创新性地提出基于安全状态的物联网设备测试方法，即为物联网设备建立安全与非安全状态，然后，根据已知的常见攻击去测试设备，看设备是否会从安全状态转化为非正常状态从而发现安全问题。但其只将该方法在智能电表上进行了测试，而且该模型的效果过度依赖于已知的攻击，无法检测出更深层次的未知漏洞。

概括而言，现阶段物联网操作系统安全测试与漏洞挖掘方法主要存在 3 个问题亟待解决。

1) 现阶段的物联网系统测试方法适用范围有限，仅仅适用于单一应用场景或系统。

2) 现有的安全测试与漏洞挖掘方法并不全面，仅仅从设备自身入手没有考虑到物联网设备相互之间的影响，缺乏广泛的实际应用的测试。

3) 目前的安全测试与漏洞挖掘方法过于单一，大多只依靠静态测试或依赖于已知攻击或常见漏洞的检测，缺乏多种测试方法综合使用以及系统运行时动态测试的方案。

3.3 物联网操作系统攻击防御

随着深度学习和大数据时代的到来，攻击者的能力不断提高，攻击手段也更多样化，物联网操作系统在未来会面临更加严重的已知和未知的系统攻击。目前，学术界主要通过在物联网设备中改进嵌入式系统异常行为检测、轻量级抵御系统攻击防御策略、安全隔离存储与数据加密及远程可信证明这 4 种安全机制来应对各种攻击手段。

3.3.1 系统异常行为检测

物联网操作系统首先应该具备异常行为检测的能力，才可以采取进一步的防御与解决措施。物联网异常行为检测与之前异常行为检测的主要不同在于物联网程序因设备的不同导致其功能差异更大，很难设计出固定的特征检测方法。

为应对物联网这一新特性，研究人员从程序自身入手，通过自动学习正常程序的特征从而检测异常行为。例如，Khan 等[41]提出动态运行时的安全监测方案，可通过检查程序运行行为和预定义行为模式的一致性来侦测攻击的发生，该方案避免了传统异常检测只能检测固定属性阈值的缺点，适用于检测未知的物联网系统攻击行为。Yoon 等[42]提出通过系统调用频率来检测异常程序行为的方法，其可以自动学习记录正常应用程序系统调用的频率分布，从而对比发现程序异常的调用行为。该方法可自动适用于各种不同的应用程序，并且其额外系统开销较低，在物联网设备中有很好的应用前景。

3.3.2 轻量级系统防御措施

由于大多数物联网设备为传感器等微型嵌入式设备，其软硬件资源均十分有限，只能执行少量的专用计算任务，没有足够的资源用于实现抵御系统攻击的防御措施。所以现阶段研究人员主要从软件、硬件 2 个方面轻量化改进原有系统防御技术，使其适用于轻量级物联网操作系统。例如，针对 ROP 攻击，加拿大多伦多大学研究人员结合 Intel MPX 硬件内存保护扩展设计了轻量级的内存保护系统，防止对关键内存区域函数调用堆栈返回地址的修改[43]，实现了比控制流完整性验证（CFI, control flow integrity）更高的安全性，同时大大降低了系统开销。还有研究人员采用建立内存影子的方法轻量化原有内存检查点设计方法，可以有效抵御基于轻量级 Linux 系统的缓冲区溢出等系统攻击[44]。

但现阶段系统针对攻击的防御技术研究大多忽视了物联网设备互用性的特点。在物联网环境下设备间的互用和依赖关系会越来越多，所以仅仅考虑抵御对自身系统的攻击是远远不够的。例如，市场上有些智能窗户控制器会根据温度传感器收集的室温自动打开或关闭窗户。在上述情景下，敌手仅需控制温度传感器的温度值，从而间

接实现对智能窗户的控制。Yu 等[45]提出基于设备间依赖关系来建立入侵检测模型，为解决设备互用问题提供了很好的解决思路，值得国内研究人员学习和参考。

3.3.3　安全隔离存储与数据加密

目前，随着物联网可穿戴设备的发展，其与用户的联系更加紧密，物联网设备存储与使用的敏感数据逐渐增多，所以不可避免地会带来用户的隐私安全问题。为避免隐私数据的泄露，禁止未经授权的运行在不可信执行环境的程序访问设备的敏感数据，安全隔离存储与数据对于物联网设备显得十分必要。

同样，由于物联网设备硬件资源十分有限，通过增加额外的安全芯片进行安全存储的方法对于物联网设备而言开销过大。针对这一问题，一方面研究人员提出了软件层面的可动态配置的安全存储策略。即在系统启动阶段，允许用户自定义地将存储器划分为安全存储和非安全存储区域，并记录对应区域的访问控制条件；然后在程序运行阶段，将内存访问指令与记录的访问控制条件进行比对，只允许受保护的安全程序访问安全存储，非安全程序不得访问安全存储中的数据[46~48]。

另一方面，研究人员从实现和设计这 2 个角度，轻量化数据加密方案来使其适用于物联网设备安全存储[49~52]，例如，通过改进 S-box 的实现方案轻量化现有加密系统[51]以及设计适用于轻量级物联网设备上的 AAβ 非对称加密方案[52]等。

但现阶段的大多数轻量化密码学方案只注重减少对设备计算与存储资源的使用，缺乏对算法耗电量的评估。而过高的电量消耗会大大降低这些算法的实用价值。所以相关研究人员在设计和实现这些轻量级安全算法时，还需充分考虑对设备电量的消耗。

3.3.4　远程可信证明

由于越来越多的小型物联网设备（如嵌入式医疗设备、特殊环境的工业控制系统以及军用设备等）在实际应用中会面临长期物理不可接触的问题，从而导致这些设备被攻击者恶意控制以后，其管理者并没有办法察觉。所以如何远程验证这些设备的关键操作是否可信成为现阶段物联网系统安全研究的热点问题。

远程可信证明是目前解决这一问题最主要也是最有效的方法之一。远程证明一般是对关键安全程序[53~56]进行验证。主要过程是发送端首先根据需要验证程序的状态信息或控制流的关键属性计算出摘要信息。然后再利用 TCB 存储的设备私钥对摘要信息进行加密。接收端也用与发送端同样的方法计算出原始程序的摘要信息。最后接收端再用发送端的公钥对加密的摘要信息进行解密从而完成对远程程序的可信证明。

但现有远程证明方法在物联网设备中的应用主要存在 2 个问题。

1) 验证过程摘要信息会不可避免地泄露程序的状态信息。为了解决敏感程序状态信息泄露的问题，有研究人员提出了基于软件属性的可信认证方法，即对软件原始属性都建立对应的安全证书，在加载软件时对每个属性证书进行验证[57,58]。然而，如何确定和提取软件的属性还是现有研究中的一大难点。

2) 另一个问题是可信证明在实现过程中会占用过多的系统资源，并不适用于轻量级物联网设备系统。所以研究人员提出只验证部分的关键安全服务程序，然后常规应用程序再利用这些安全服务程序进行可信安全操作[59]，从而简化远程可信证明过程。同时，最近许多研究人员尝试结合物联网设备自身独特的物理特性（PUF, physical unclonable function）来辅助认证过程[60~62]。即 PUF 在认证过程中生成动态的“挑战—响应”对来进行验证，从而代替存储固定的设备密钥，节约用于单独存储密码的硬件资源，同时提高了安全性[62]。然而该方案也存在一定的安全问题，例如，挑战响应对不能被重复利用，否则，会导致重放攻击。所以目前更加实用的方案是将认证密钥存储在基于 PUF 的密钥存储器中[63,64]。

3.4　不同场景对应的安全技术

为了最大限度地保证其安全性，物联网应用应该具备“系统安全构建—系统安全性分析—系统攻击防御”这整个周期内的所有安全技术。不过根据 2.3 节介绍的不同场景的安全需求，不同的应用场景下的操作系统对安全技术要求的侧重点有所不同，如表 3 所示。

表 3　不同场景下物联网操作系统采取的安全技术

场景	技术
智能家居	安全存储与数据加密、平台组件完整性验证
智能医疗	安全存储与数据加密、可信隔离的执行环境、远程可信证明
智能工业	可信隔离的执行环境、异常行为检测、安全接口、远程可信证明
智能汽车	安全内核、可信隔离的执行环境、平台组件完整性验证

4 物联网操作系统的挑战与机遇

在深入调研现阶段物联网操作系统安全问题的基础上，指出物联网操作系统安全研究中目前面临的挑战。然后，结合研究现状给出可用于应对这些挑战的安全技术机遇，其对应关系如表4所示。

表4 物联网操作系统面临的挑战和机遇

挑战	机遇
不安全的系统构建	安全系统框架设计
设备资源的有限性	轻量化系统防御措施
不可接触的物理设备	远程可信认证
存在漏洞的系统	系统安全测试与漏洞检测方法
隐私数据泄露	轻量化安全存储
外围设备安全威胁	安全接口设计
关键程序入侵	安全内核、可信隔离的执行环境
各种系统攻击	异常行为检测、可信隔离的执行环境、平台组件完整性验证

4.1 不安全的系统构建

目前，物联网操作系统安全问题产生的根本原因主要是在系统构建时忽略了安全因素。但小型厂商并不具备安全系统构建的专业知识，所以需要安全研究人员设计出实用的且额外成本低的安全系统构建框架[65]供物联网设备厂商选择使用。另外，研究人员可设计额外的安全评估模块在系统设计过程中就预先对其进行安全性分析[66]，防患于未然。

4.2 设备资源的有限性

由于物联网设备的计算、存储资源有限，并对设备的成本和功耗有着较高的要求，所以在保证操作系统安全的同时还要使附加的安全机制的功耗和资源使用降到最低，才能切实提高安全机制的实用价值。现有轻量加密算法[67]、轻量认证算法[68]以及轻量级系统防御措施[69]的资源消耗和安全性均还无法满足现阶段轻量级物联网设备的安全需求。

4.3 不可接触的物理设备

物联网环境下很多设备都会面临长期物理不可接触，如嵌入式医疗设备、特殊环境的工业控制系统和军用设备等。如何验证这些设备关键操作的可信性以及数据的可靠性逐渐成为现阶段物联网操作系统研究的一大热点，需要研究人员提出更加轻量化且高效的远程可信认证方案来解决这一难题[70]。

4.4 存在漏洞的系统

现阶段物联网应用的操作系统中存在大量安全漏洞，但现有的物联网安全测试工具与漏洞挖掘方法过于简单或直接照搬原有Android系统的测试方法，无法挖掘出更加深入的物联网系统中的安全问题，同时发现的安全问题也不够全面。根据本文调研结果，目前尚未发现优秀的针对物联网系统安全测试的公开成果，亟待研究人员提出更加有效的安全分析与测试工具[30,71]。

4.5 隐私数据泄露

随着物联网设备越发普及，智能家居、智能医疗设备等还会收集用户大量的隐私信息，如室温变化、体征变化等，保管传输不当会导致严重的用户隐私泄露问题。但这些物联网设备存储资源均十分有限，系统防御能力十分薄弱，如何在轻量级物联网设备系统中利用更少的系统资源构建出可信安全的存储空间防止隐私数据泄露，需要引起研究人员的重点关注。

4.6 外围设备安全威胁

目前，物联网设备之间的无线与有线交互越来越频繁，仅仅保障设备内部系统安全往往是不够的。在物联网系统设计中还需要特别对外围接口的程序设计以及调用进行仔细的检查。防止攻击者利用不安全的外围设备入侵关键设备系统。设计出安全、灵活、广泛适用的程序接口也是现阶段物联网操作系统安全研究中不可忽视的环节[72]。

4.7 关键程序入侵

随着物联网设备在工业等关键设施中的广泛应用，其安全问题也越发严重，攻击者可以通过入侵控制基础设备的关键程序从而造成严重的物理破坏。故对于控制重要设备的关键程序，一方面需要为其构造可信隔离的安全执行环境，即使在操作系统被攻破的前提下仍然保障关键程序不会受到威胁；另一方面需要构建安全内核，增加操作系统抵御攻击的能力。

4.8 各种系统攻击

由于物联网系统普遍存在诸多漏洞，并且随着攻击者能力不断提高，物联网操作系统随时可能遭受多种类型的系统攻击。对此，需要从各个方面构建起完整的系统防御机制。首先，要从启动阶段就开始对系统进行防护，为关键程序构建可信执行环境进行隔离[44]；其次，实时对物联网操作系统上程序行为进行监控，及时发现和处理异常行为，有效

提高设备的安全性[50,51]；最后，针对严重的特定攻击手段（如精心设计的 ROP 攻击、侧信道分析等），设计出对应的高效防御措施[32,73]。

5　未来研究方向展望

根据第 4 节介绍的物联网挑战与机遇，本文将其中需要进一步发展的研究技术提炼抽象为未来研究的发展方向，8 个技术发展机遇关系如图 3 所示，并在此之后根据现有系统防御技术的不足，提出了物联网生存技术这一新的研究方向。

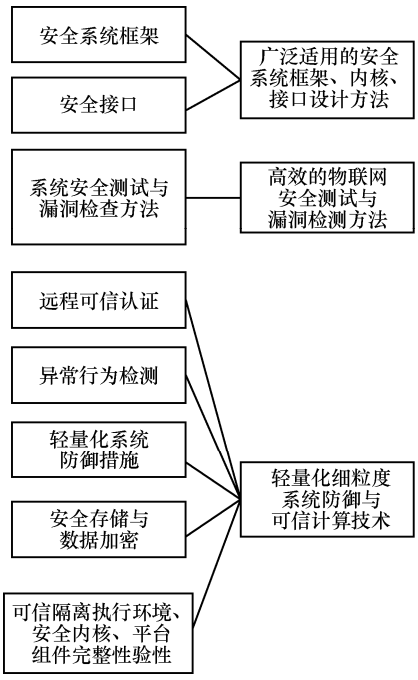

图 3　物联网操作系统研究方向与现有嵌入式系统安全防御技术关系

5.1　轻量化细粒度系统防御与可信计算技术

现阶段许多传感器和小型物联网设备软硬件资源均十分有限，操作系统也十分轻量，并不具备如 DEP、ASLR 等普通计算机的系统防御措施，甚至硬件架构也不支持 MMU 等内存管理功能。但如果依靠增加外围硬件如安全芯片来实现可信计算，开销过大且不易推广。为了解决细粒度的系统保护与资源有限的矛盾，需要充分利用现有设备的软硬件资源例如 ARM 架构的 MPU 和 TrustZone 等，构建适用范围更广的轻量级系统防御与可信计算技术。

5.2　广泛适用的安全系统框架、内核、接口设计方法

现阶段物联网设备种类越来越多，并逐步应用于医疗、家居、交通和工业等各种不同的场景，所以设备间软硬件架构普遍存在异构性。但是，对每种设备都定制化构建安全系统又是不切实际的。如何对这些异构设备设计出广泛适用的安全系统构建方法、安全内核及外围接口将成为物联网系统安全研究的一大难点。

5.3　高效的物联网安全测试与漏洞检测方法

现阶段，各种物联网设备和系统层出不穷，许多未经严格安全测试存在大量安全漏洞的物联网产品已经流入市场。现有的物联网设备测试方法并不成熟，缺乏大量的产品测试[36]，而且测试方法也过于简单，无法挖掘出深层次的安全漏洞，例如，设备间互用导致的安全问题。如何对各种物联网产品进行深入全面的系统测试和漏洞检测逐步成为物联网操作系统安全研究领域亟待解决的一大问题。

5.4　物联网系统生存技术

在对物联网操作系统安全研究现状进行深入分析时发现，现阶段物联网操作系统还没有关于系统生存技术的相关研究。而随着物联网设备种类与功能的不断增加，物联网操作系统也会更加复杂，移除所有存在于物联网系统中可能被攻击者利用的漏洞是一件十分困难的事情，因此，本文认为研究生存技术在物联网操作系统中的应用十分必要。而入侵容忍系统能够在即使部分组件被妥协的状况下，保证整个系统仍发挥正常的功能。关于传统的系统生存技术，目前已有许多研究成果[74~76]。例如，对当前操作系统进行自动评估，并帮助用户选择和配置相应的生存与入侵容忍机制[74]以及为入侵行为构建状态转换模型用户自动学习入侵特征等[75]。但这些技术过于复杂还无法直接应用于物联网系统中，目前尚未发现适用于物联网系统的生存技术，亟待相关研究人员填补这项空白。

6　结束语

本文首先介绍了物联网系统架构及其特征并与传统的嵌入式系统进行了比较，然后在调研了大量相关文献后，创新性地从“系统安全构建—系统安全性分析—系统攻击防御”的角度对现有物联网操作系统相关研究进行分类总结。进一步在此基础上指出了物联网操作系统安全面临的 8 个挑战与机遇，最后，对未来物联网操作系统安全研究方向进行了展望，指出轻量化细粒度系统防御与可信计算技术、广泛适用的安全系统框架、内核、接口设计方法、高效的物联网安全测试与漏洞检测方法、物联网系统生存技术等这些物联网操作系统安全的

未来热点研究方向。

物联网操作系统是物联网发展的重要基础，只有保证了物联网操作系统的安全，才能进一步保证物联网的安全，促进物联网产业快速发展与普及，更好地服务于人们的日常生活。

参考文献：

[1] 张玉清, 周威, 彭安妮. 物联网安全综述[J]. 计算机研究与发展, 2017, 54(10):2130-2143.
ZHANG Y Q, ZHOU W, PENG A N. Survey of Internet of things security[J]. Journal of Computer Research and Development, 2017, 54(10): 2130-2143.

[2] AMIRI-KORDESTANI M, BOURDOUCEN H. A survey on embedded open source system software for the Internet of things[C]// Free and Open Source Software Conference. 2017.

[3] LANGNER R. Stuxnet: dissecting a cyberwarfare weapon[J]. IEEE Security & Privacy, 2011, 9(3):49-51.

[4] D'EXPLOITATION S. RIOT-the friendly operating system for the Internet of Things-VIDEO[J]. Genomics & Informatics, 2012, 10(4): 249-55.

[5] DUNKELS A, GRNVALL B, VOIGT T. Contiki-a lightweight and flexible operating system for tiny networked sensors[C]// IEEE International Conference on Local Computer Networks. 2004:455-462.

[6] PAVELIĆ N. Evaluation of Android things platform[D].Sveučilište u Zagrebu: Fakultet Elektrotehnike i Računarstva, 2017.

[7] TOULSON R, WILMSHURST T. Fast and effective embedded systems design: applying the ARM mbed[J]. Newnes, 2016.

[8] SHALAN M, EL-SISSY D. Online power management using DVFS for RTOS[C]//4th International Design and Test Workshop (IDT). 2009: 1-6.

[9] INAM R, MÄKI-TURJA J, SJÖDIN M, et al. Hard real-time support for hierarchical scheduling in FreeRTOS[C]//23rd Euromicro Conference on Real-Time Systems. 2011: 51-60.

[10] CAO Q, ABDELZAHER T, STANKOVIC J, et al. The liteos operating system: towards unix-like abstractions for wireless sensor networks[C]//International Conference on Information Processing in Sensor Networks. 2008: 233-244.

[11] GRÄS S, LOSE G. Green hills software's integrity real-time operating system unleashes the power of Intel network processors[J]. International Urogynecology Journal, 2013, 24(10):1771.

[12] POELLABAUER C, SCHWAN K, WEST R, et al. Flexible user/kernel com-munication for real-time applications in elinux[C]//The Workshop on Real Time Operating Systems and Applications and Second Real Time Linux Workshop (in conjunction with RTSS 2000). 2000.

[13] VELEZ G, SENDEROS O, NIETO M, et al. Implementation of a computer vision based advanced driver assistance system in Tizen IVI[C]// ITS World Congress. 2014.

[14] ZHAO K, GE L. A survey on the Internet of things security[C]// Ninth International Conference on Computational Intelligence and Security. 2013:663-667.

[15] ZARAGOZA M G, KIM H K, LEE R Y. Big data and IoT for *u*-healthcare security[M]//Computer and Information Science. Springer International Publishing, 2018:1-11.

[16] HENRY N L, PAUL N R, MCFARLANE N. Using bowel sounds to create a forensically-aware insulin pump system[C]//Usenix Conference on Safety, Security, Privacy and Interoperability of Health Information Technologies. 2013: 8.

[17] LANGNER R. Stuxnet: dissecting a cyberwarfare weapon[J]. IEEE Security & Privacy, 2011, 9(3):49-51.

[18] CLARK S S, RANSFORD B, RAHMATI A, et al. WattsUpDoc: power side channels to nonintrusively discover untargeted malware on embedded medical devices[C]//HealthTech. 2013.

[19] WOO S, JO H J, LEE D H. A practical wireless attack on the connected car and security protocol for in-vehicle CAN[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(2): 993-1006.

[20] HUMAYED A, LUO B. Cyber-physical security for smart cars: taxonomy of vulnerabilities, threats, and attacks[C]//The ACM/IEEE Sixth International Conference on Cyber-Physical Systems. 2015: 252-253.

[21] FRANCILLON A. Analyzing thousands of firmware images and a few physical devices: what's next?[C]//The 6th International Workshop on Trustworthy Embedded Devices. 2016: 1.

[22] BABAR S, STANGO A, PRASAD N, et al. Proposed embedded security framework for Internet of things (IoT)[C]//2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronics Systems Technology (Wireless VITAE). 2011: 1-5.

[23] JIN Y. Embedded system security in smart consumer electronics[C]//The 4th International Workshop on Trustworthy Embedded Devices. 2014: 59.

[24] LIU S. Design and development of a security kernel in an embedded system[J]. International Journal of Control & Automation, 2014, 7(11):49-58.

[25] GUANCIALE, ROBERTO, KHAKPOUR, et al. Formal verification of information flow security for a simple arm-based separation kernel[J]. Journal of Molecular Structure Theochem, 2013, 587(s1-3): 49-56.

[26] AZAB A M, SWIDOWSKI K, BHUTKAR R, et al. SKEE: a lightweight secure kernel-level execution environment for ARM[C]//NDSS. 2016.

[27] BATES A, TIAN D, BUTLER K R B, et al. Trustworthy whole-system provenance for the Linux kernel[C]//Usenix Conference on Security Symposium. 2015: 319-334.

[28] MALENKO M, BAUNACH M. Real-time and security requirements for Internet-of-things operating systems[C]//Internet Der Dinge: Echtzeit 2016. 2016: 33-42.

[29] DYER J G, LINDEMANN M, PEREZ R, et al. Building the IBM 4758 secure coprocessor[J]. Computer, 2001, 34(10): 57-66.

[30] PETRONI JR N L, FRASER T, MOLINA J, et al. Copilot-a coprocessor-based kernel runtime integrity monitor[C]//USENIX Security Symposium. 2004: 179-194.

[31] ZHAO L, LI G, SUTTER B D, et al. ARMor: fully verified software fault isolation[C]//The International Conference on Embedded Software. 2011:289-298.

[32] CHEN X, GARFINKEL T, LEWIS E C, et al. Overshadow:a virtualization based approach to retrofitting protection in commodity operating systems[C]//ACM, 2008:2-13.

[33] NORDHOLZ J, VETTER J, PETER M, et al. Xnpro: low-impact hypervisor-based execution prevention on ARM[C]//The 5th International Workshop on Trustworthy Embedded Devices. 2015: 55-64.

[34] PARK D J, HWANG H S, KANG M H, et al. Secure boot method and semiconductor memory system using the method: US20090019275[P]. 2009.

[35] KIRKPATRICK M S, GHINITA G, BERTINO E. Resilient authenticated execution of critical applications in untrusted environments[J]. IEEE Transactions on Dependable & Secure Computing, 2012, 9(4):597-609.

[36] KOHNHÄUSER F, KATZENBEISSER S. Secure code updates for mesh networked commodity low-end embedded devices[C]//European Symposium on Research in Computer Security. 2016: 320-338.

[37] COSTIN A, ZADDACH J, FRANCILLON A, et al. A large-scale analysis of the security of embedded firmwares[C]//USENIX Security Symposium. 2014: 95-110.

[38] SACHIDANANDA V, TOH J, SIBONI S, et al. POSTER: towards exposing Internet of things: a roadmap[C]//ACM Sigsac Conference on Computer and Communications Security. 2016:1820-1822.

[39] MER M, ASPINALL D, WOLTERS M. POSTER: weighing in eHealth security[C]//ACM Sigsac Conference on Computer and Communications Security. 2016:1832-1834.

[40] TABRIZI F M, PATTABIRAMAN K. Formal security analysis of smart embedded systems[C]//The 32nd Annual Conference on Computer Security Applications. 2016: 1-15.

[41] KHAN M T, SERPANOS D, SHROBE H. A rigorous and efficient run-time security monitor for real-time critical embedded system applications[C]//2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). 2016: 100-105.

[42] YOON M K, MOHAN S, CHOI J, et al. Learning execution contexts from system call distribution for anomaly detection in smart embedded system[C]//2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI). 2017: 191-196.

[43] HUANG W, HUANG Z, MIYANI D, et al. LMP: light-weighted memory protection with hardware assistance[C]//The 32nd Annual Conference on Computer Security Applications. 2016: 460-470.

[44] VOGT D, GIUFFRIDA C, BOS H, et al. Lightweight memory checkpointing[C]//IEEE/IFIP International Conference on Dependable Systems and Networks. 2015:474-484.

[45] YU T, SEKAR V, SESHAN S, et al. Handling a trillion (unfixable) flaws on a billion devices: rethinking network security for the Internet-of-things[C]//ACM Workshop on Hot Topics in Networks. 2015:5.

[46] KOEBERL P, SCHULZ S, SADEGHI A R, et al. TrustLite: a security architecture for tiny embedded devices[C]//European Conference on Computer Systems. 2014:10.

[47] DEFRAWY K E, PERITO D, TSUDIK G. SMART: secure and minimal architecture for (Establishing a Dynamic) root of trust[J]. Isoc, 2017.

[48] STRACKX R, PIESSENS F, PRENEEL B. Efficient isolation of trusted subsystems in embedded systems[C]//International Conference on Security and Privacy in Communication Systems. 2010:344-361.

[49] GUO F, MU Y, SUSILO W, et al. CP-ABE with constant-size keys for lightweight devices[J]. IEEE Transactions on Information Forensics &Security, 2014, 9(5):763-771.

[50] SHI Y, WEI W, HE Z, et al. An ultra-lightweight white-box encryption scheme for securing resource-constrained IoT devices[C]//Conference on Computer Security Applications. 2016:16-29.

[51] BANSOD G, RAVAL N, PISHAROTY N. Implementation of a new lightweight encryption design for embedded security[J].IEEE Transactions on Information Forensics and Security, 2015, 10(1): 142-151.

[52] ADNAN S F S, ISA M A M, HASHIM H. Timing analysis of the lightweight AAβ encryption scheme on embedded Linux for Internet of things[C]//2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). 2016: 113-116.

[53] KAUER B. OSLO: improving the security of trusted computing[C]// USENIX Security Symposium. 2007: 229-237.

[54] KÜHN U, SELHORST M, STÜBLE C. Realizing property-based attestation and sealing with commonly available hard and software[C]//The 2007 ACM workshop on Scalable trusted computing. 2007: 50-57.

[55] KYLÄNPÄÄ M, RANTALA A. Remote attestation for embedded systems[C]//Conference on Cybersecurity of Industrial Control Systems. 2015: 79-92.

[56] TSUDIK G. Challenges in remote attestation of low-end embedded devices[C]//The 4th International Workshop on Trustworthy Embedded Devices. 2014: 1.

[57] CHEN L, LÖHR H, MANULIS M, et al. Property-based attestation without a trusted third party[J]. Information Security, 2008: 31-46.

[58] SADEGHI A R, STÜBLE C. Property-based attestation for computing platforms: caring about properties, not mechanisms[C]//The 2004 workshop on new security paradigms. 2004: 67-77.

[59] MCCUNE J M, LI Y, QU N, et al. TrustVisor: efficient TCB reduction and attestation[C]//2010 IEEE Symposium on Security and Privacy (SP). 2010: 143-158.

[60] SCHULZ S, WACHSMANN C, SADEGHIS A R. Lightweight remote attestation using physical functions, technische universitat darmstadt, darmstadt[R]. Germany, Technical Report, 2011.

[61] SCHULZ S, SADEGHI A R, WACHSMANN C. Short paper: lightweight remote attestation using physical functions[C]//The fourth ACM Conference on Wireless Network Security. 2011: 109-114.

[62] RANASINGHE D, ENGELS D, COLE P. Security and privacy: modest proposals for low-cost RFID systems[C]//Auto-ID Labs Research Workshop, Zurich, Switzerland. 2004.

[63] EICHHORN I, LEEST V V D, LEEST V V D. Logically reconfigurable PUFs: memory-based secure key storage[C]//ACM Workshop on Scalable Trusted Computing. 2011:59-64.

[64] YU M D M, M'RAIHI D, SOWELL R, et al. Lightweight and secure PUF key storage using limits of machine learning[C]//International Work-shop on Cryptographic Hardware and Embedded Systems. 2011: 358-373.

[65] GARITANO I, FAYYAD S, NOLL J. Multi-metrics approach for

security, privacy and dependability in embedded systems[J]. Wireless Personal Communications, 2015, 81(4): 1359-1376.

[66] OH D, KIM D, RO W W. A malicious pattern detection engine for embedded security systems in the Internet of things[J]. Sensors, 2014, 14(12): 24188-24211.

[67] BANSOD G, RAVAL N, PISHAROTY N. Implementation of a new lightweight encryption design for embedded security[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(1): 142-151.

[68] ODELU V, DAS A K, GOSWAMI A. A secure biometrics-based multi-server authentication protocol using smart cards[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(9): 1953-1966.

[69] CARABAS M, MOGOSANU L, DEACONESCU R, et al. Lightweight display virtualization for mobile devices[C]//International Workshop on Secure Internet of Things. 2014:18-25.

[70] ABERA T, ASOKAN N, DAVI L, et al. C-FLAT: control-flow attestation for embedded systems software[C]//The 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 743-754.

[71] CHALUPAR G, PEHERSTORFER S, POLL E, et al. Automated reverse engineering using Lego[J]. WOOT, 2014, 14: 1-10.

[72] ASOKAN N, EKBERG J E, KOSTIAINEN K, et al. Mobile trusted computing[J]. Proceedings of the IEEE, 2014, 102(8):1189-1206.

[73] HALEVI T, MA D, SAXENA N, et al. Secure proximity detection for NFC devices based on ambient sensor data[C]//European Symposium on Research in Computer Security. 2012: 379-396.

[74] LIN J, JING J, LIU P. Evaluating intrusion-tolerant certification authority systems[J]. Quality & Reliability Engineering International, 2012, 28(8):825-841.

[75] GOSEVAPOPSTOJANOVA K, VAIDYANATHAN K, TRIVEDI K, et al. Characterizing intrusion tolerant systems using a state transition model[C]//DARPA Information Survivability Conference & Exposition II. 2001:211-221.

[76] GUPTA V, LAM V, RAMASAMY H G V, et al. dependability and performance evaluation of intrusion-tolerant server architectures[M]// Dependable Computing. Springer Berlin Heidelberg, 2003: 81-101.

[作者简介]

彭安妮（1995-），女，湖北武汉人，中国科学院大学博士生，主要研究方向为网络与系统安全。

周威（1993-），男，河北保定人，中国科学院大学博士生，主要研究方向为网络与系统安全。

贾岩（1992-），男，河北石家庄人，西安电子科技大学博士生，主要研究方向为网络与系统安全。

张玉清（1966-），男，陕西宝鸡人，博士，中国科学院大学教授，主要研究方向为网络与信息系统安全。