中国通信标准化协会
China Communications Standards Association

物联网操作系统安全白皮书
（2022 年）

中国通信标准化协会

2022年9月

版权说明

前　言

物联网操作系统是指运行在物联网感知控制域中各类终端上的系统软件，主要实现对物理世界对象的本地化感知、协同和操控，并为物联网其他域提供远程管理和服务接口，是感知控制域中各类终端的主要功能的载体。因此物联网操作系统的稳定和安全是物联网终端以及物联网整体系统的安全基础。目前，物联网操作系统种类繁多，但普遍存在安全能力参差不齐，安全设计缺失或不规范，安全防护能力不足等问题，导致物联网终端设备成为了物联网系统中的安全薄弱环节。

在深入分析物联网操作系统所面临的主要安全问题和风险的基础上，本白皮书旨在指出能有效保护物联网操作系统的安全机制、安全体系以及安全技术，推动适合于物联网设备及操作系统的安全技术的研发和应用。希望能够为产业在规划设计物联网操作系统安全能力时提供参考和指引。

本白皮书由中国通信标准化协会（CCSA）网络与信息安全（TC8）安全基础工作组（WG4）牵头，参与编写单位包括：大唐高鸿信安（浙江）信息科技有限公司、中国网络安全审查技术与认证中心、中国移动通信有限公司研究院、中移物联网有限公司、中兴通讯股份有限公司、公安部第三研究所、元心信息科技集团有限公司、北京邮电大学、北京梆梆安全科技有限公司、北京豆荚科技有限公司、电子科技大学、安谋科技（中国）有限公司、美的集团股份有限公司、杭州安恒信息技术股份有限公司、南京翼辉信息技术有限公司、烽火通信科技股份有限公司、深圳大学、绿盟科技集团股份有限公司、意法半导体（中国）投资有限公司。主要参与编写人员：王亚鑫、王伟、王雷、王聪、王羲文、卢延波、卢佐华、申永波、田丽丹、李东宏、李蒙、李孝成、刘尚焱、刘军、刘海洁、刘国锋、刘伟丽、许睿、牟飞、陈丽蓉、陈珊、何申、何狄凡、吴国燕、杨辉、杨坤、杨明、杨新苗、余希希、张亮亮、邹仕洪、罗蕾、国炜、郑驰、郝卓航、奚智、徐祥智、袁森、阎军智、黄静、蒋学鑫、彭凯、粟栗、路晔绵、魏凡星。

目 录

1. 物联网操作系统概述

1.1. 物联网及物联网操作系统

1.1.1. 物联网简介及发展趋势

物联网是“通过感知设备，按照既定协议，连接物、人、系统和信息资源， 对物理和虚拟世界的信息进行处理并做出反应的智能服务系统”。其中，“物”指物理实体。国际标准 ISO/IEC 22417:2017《 Internet of things (IoT) - IoT use cases》中提出物联网的应用场景包括交通、家居、公共建筑、办公、工业、农业、渔业、穿戴、机车、智慧城市等。

全球物联网连接数保持高速增长，2020 年全球物联网总连接数达到 131 亿，预计到 2025 年，连接规模将达到 246 亿，全球物联网行业正处于高速发展期。我国物联网连接数在全球占比超过 30%，产业规模突破 1.7 万亿元[1]，呈现出良好的增长态势。

2021 年，工信部发布《物联网新型基础设施建设三年行动计划》，明确提出“融合应用发展行动”，在社会治理领域，将感知终端纳入公共基础设施建设，加快构建智慧城市、数字乡村。各地政府将其纳入新阶段发展重点，物联网投资将持续加大，外部政策为其快速发展注入了新动力。

1.1.2. 物联网操作系统简介及架构

物联网操作系统是支持物联网技术大规模发展的核心基础软件，包括操作系统内核、外围组件和服务、物联网安全框架等，以支持构成具有低功耗、安全通信属性的物联网软件平台。物联网操作系统的内核通常具备任务管理、中断管理、异常处理、时钟管理、存储管理、同步与通信等功能。物联网操作系统向下协调和控制各种软件硬件资源，向上提供统一的应用编程接口，降低物联网应用开发的复杂度、成本和时间。物联网操作系统的架构如图 1 所示。

由于物联网操作系统具有不同的架构和安全机制，因此在安全性、AI 支持、实时性、资源要求等方面具有较大的差异。如工业控制的物联网操作系统必须满足强实时性要求，而用于智能家居终端设备的物联网操作系统的实时性要求则不高。

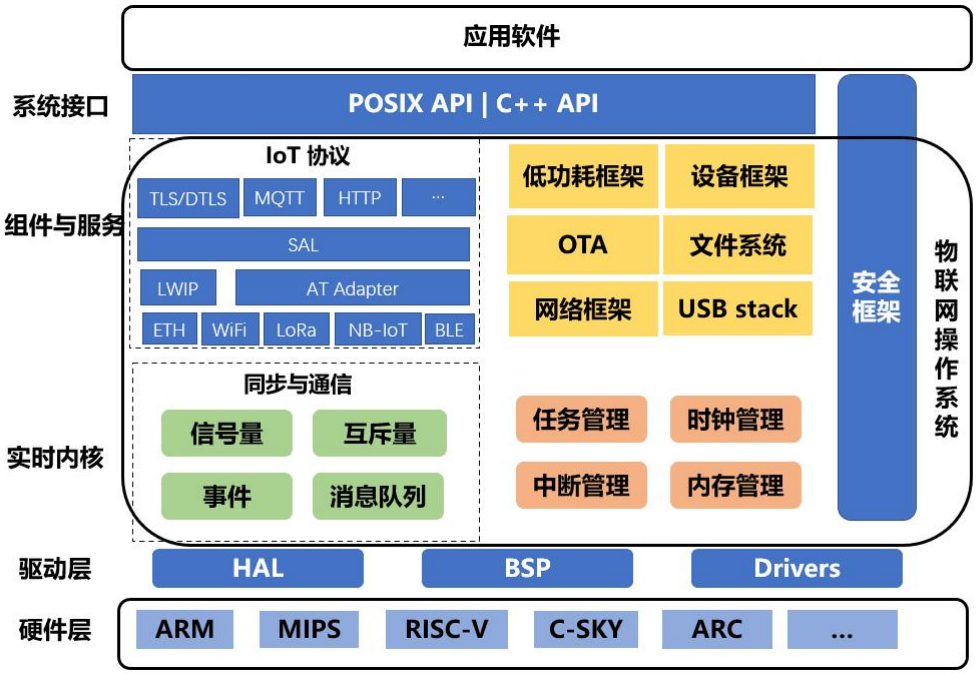

图 1 物联网操作系统架构示意图

1.1.3. 物联网操作系统特点

近些年，随着设备形态多样化发展，特别是人机交互方式的更迭，传统操作系统逐步从企业商用操作系统、个人计算机操作系统演进到移动设备操作系统。

企业商用机领域的大型机、小型机的操作系统以类 UNIX 系统为主，而人机交互方式以键盘为主。如果对于磁盘 I/O 要求较高，那么 Linux 是首选服务器操作系统。

个人计算机中微软的 Windows 占据主流，苹果的 macOS 独树一帜，Linux 各种发行版后来居上。人机交互方式演进为鼠标、键盘为主。图形用户界面友好，窗口制作优美，操作简单易学。

移动设备的操作系统目前以谷歌 Android 系统和苹果 iOS 系统为主。由于设备多是便携的小尺寸手持形态，人机交互方式取消了键盘，演进为完全基于触摸屏的交互设计。

相比传统操作系统，物联网操作系统通常具备如下的特性[2]：

1、可裁剪伸缩性：根据不同的硬件能力和应用场景，需要对物联网操作系统进行剪裁与配置，以灵活的配置来满足不同的需求。操作系统要实现上述的灵活配置，需要采用“组件化、模块化”的思想，如可伸缩的开放式架构、组件的模块化设计以及任务调度分层化管理等。

2、低功耗节能性：由于部署的位置、空间、热环境等方面的限制，低功耗成为物联网设备及操作系统的一个非常关键的指标。在物联网操作系统整体架构设计的时候，加入一些休眠模式、节能模式、降频模式等逻辑判断，以支持足够

的电源续航能力。

3、安全性：作为物联网基础软件的操作系统，尤其需要重视信息安全性，具备防御外部入侵和避免非授权访问的能力。其次需要重视功能安全性，避免因软件功能缺陷而导致安全风险。

4、实时性：物联网应用领域中大部分设备都要求实时性，不论是数据采集、信息交互还是操作控制。实时操作系统（RTOS）是指当外界事件或数据产生时，能够实时采集并以足够快的速度予以处理，其处理的结果又能在规定的时间之内来控制生产过程或对处理系统做出快速响应，调度一切可利用的资源完成实时任务，并控制所有实时任务协调一致运行的操作系统。

5、泛在通信性：由于物联网设备的部署场景千差万别，可能采用的通信制式五花八门，因此作为物联网基础软件的操作系统，就要求内置各种近距离和远距离的通信协议，既能支持 GPRS/HSPA/4G/5G/NB-IoT 等蜂窝无线通信功能，也能支持 WiFi/ZigBee/NFC/RFID 等近场通信功能。

6、云端连接性：物联网设备完成数据采集后，海量数据通常需要云端进行存储与分析。因此，云端连接性是物联网操作系统的基础功能。通过内置云平台连接中间件，物联网操作系统可以极大的简化物联网应用的开发。

1.1.4. 物联网操作系统发展趋势

由于物联网应用场景的多样性，使得物联网终端复杂多样，为了满足不同应用的需求，物联网操作系统产品种类十分丰富。目前物联网操作系统呈现出三种主要的形态[3]。一是以谷歌 Android Wear、苹果 watchOS 为代表的操作系统，通过对智能手机操作系统或 PC 操作系统进行裁剪以适配物联网需求，但往往难以满足物联网级别的功耗和可靠性要求。二是在传统嵌入式 RTOS 上增加物联网通信功能，如 FreeRTOS、RT-Thread 等，此类操作系统具有功耗低、可靠性高等特点，但缺乏良好的应用生态。三是物联网专用操作系统，具备可伸缩、易扩展、强实时性、高可靠性等特点，可以更好地适配各类物联网的应用需求，如阿里巴巴 AliOS Things、中国移动 OneOS 等。

物联网操作系统发展成熟仍需要时间，一是由于新型物联网操作系统对主流应用软件的兼容性问题，二是物联网操作系统作为系统软件，涉及到整个生态的建设，而生态建设、应用研发适配以及开发者培育都需要时间。

1.2. 典型物联网操作系统安全架构

安全的物联网操作系统需要从系统设计、实现、使用和管理各个阶段入手，遵循一套完善的系统安全策略。物联网操作系统内核中存在错误或设计缺陷，应用部分采取再多缓解措施也难以保障系统的安全性。通过微内核设计来减少内核

的复杂度、利用安全核来提供整个物联网操作系统安全性，成为提升物联网操作系统安全性的一个趋势。

当前，物联网操作系统主要分为两大类。一类主要面向资源受限的物联网设备，系统架构多采用可配置、高度模块化的设计，编译后的内核通常小于 10KB，这类物联网操作系统普遍没有用户空间的概念，功能较为单一，常见的有μC/OS、FreeRTOS、Contiki、Mbed OS、QNX® Neutrino® RTOS、Zephyr、ThreadX、LiteOS、AliOS Things 等。另一类则面向资源丰富的物联网设备，多采用 UNIX 或类 UNIX 内核，除了提供进程调度、进程间通信等基础服务外还提供文件系统、设备驱动、虚拟内存管理、网络协议栈等复杂的服务，这类物联网操作系统功能繁多，运行环境复杂，安全问题突出，常见的有 Linux、QNX、Android、鸿蒙 OS 等。

1、资源受限型物联网操作系统

资源受限型物联网操作系统典型架构如图 2 所示，主要包括硬件层、内核层、服务层/框架层、应用层。受限于硬件性能，该类物联网操作系统安全功能较为薄弱，如何在安全与可用性之间取得平衡是这类物联网操作系统设计的重点。轻量化的操作系统安全技术是当前的主要方向，例如：Arm 在 Armv8-M 中引入了 TrustZone-M 技术，并提供了 TF-M 固件安全解决方案；翼辉提供了嵌入式防火墙，能有效防御常见的网络攻击；OneOS 提供了轻量级 TLS，可利用极低的资源消耗实现数据加密和安全通信服务；LiteOS 提供了 LMS（Lite Memory Sanitizer）服务，能够实时检测内存操作的合法性。

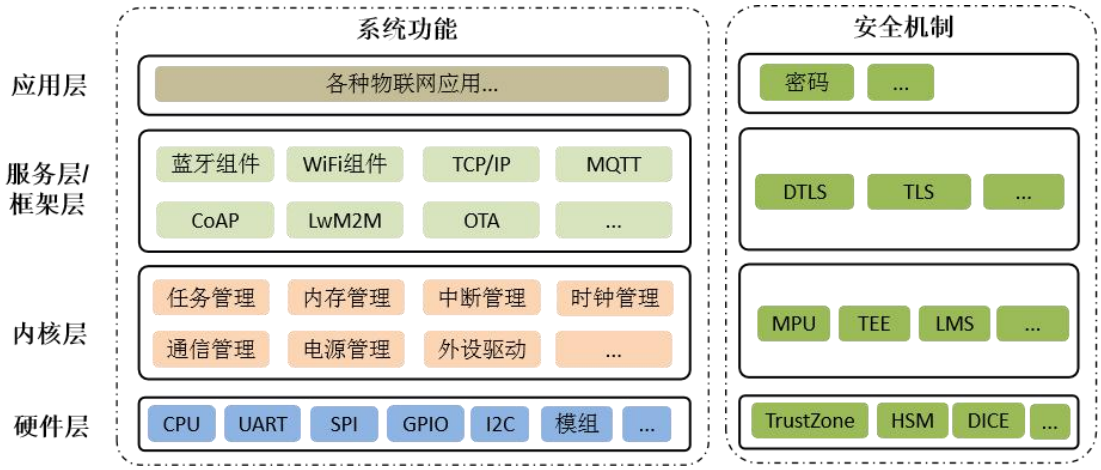

图 2 典型资源受限型物联网操作系统及安全架构

2、资源丰富型物联网操作系统

资源丰富型物联网操作系统典型架构如图 3 所示，主要包括硬件资源层、内核层、系统组件/服务/工具层、文件系统、应用层。这类物联网操作系统多采用 UNIX 或类 UNIX 内核。考虑到物联网设备的使用环境、使用方式存在较大差异，传统计算机操作系统所采用的安全模型、安全机制并不一定适用于物联网操作系统。因此，物联网操作系统在设计开发时需要结合物联网设备特性选择合适的安

全模型和安全机制来保障系统的安全性。例如：传统计算机操作系统善于保护某一个用户不受其他用户的影响，但对于物联网设备而言，基本以 root（超级管理员用户）身份运行，所以更加关注相同用户的不同进程之间的访问控制，这种情况下使用类型增强访问控制机制（Type Enforcement Access Control，TEAC）会更合适。

可选的安全模型及安全机制有：BLP 模型、Bida 模型、Clark-Wilson 模型、Chinese Wall 模型等；基于硬件的内存保护机制、运行域保护机制、I/O 保护机制；基于软件的标识与鉴别机制、访问控制机制、最小特权管理机制、可信通路机制、隐蔽通道的分析和处理、安全审计机制等。这些机制现在已经有了很多成熟的技术实现。以访问控制为例，Linux 提供了 LSM（Linux 安全模块）框架，可以很方便的实现各种访问控制模型及策略；SELinux 提供了基于角色、类型增强、多级安全的访问控制机制；简化的强制访问控制内核（Simplified Mandatory Access Control Kernel，SMACK）提供了类型增强访问控制机制；Tomoyo 提供了基于路径的访问控制机制等。

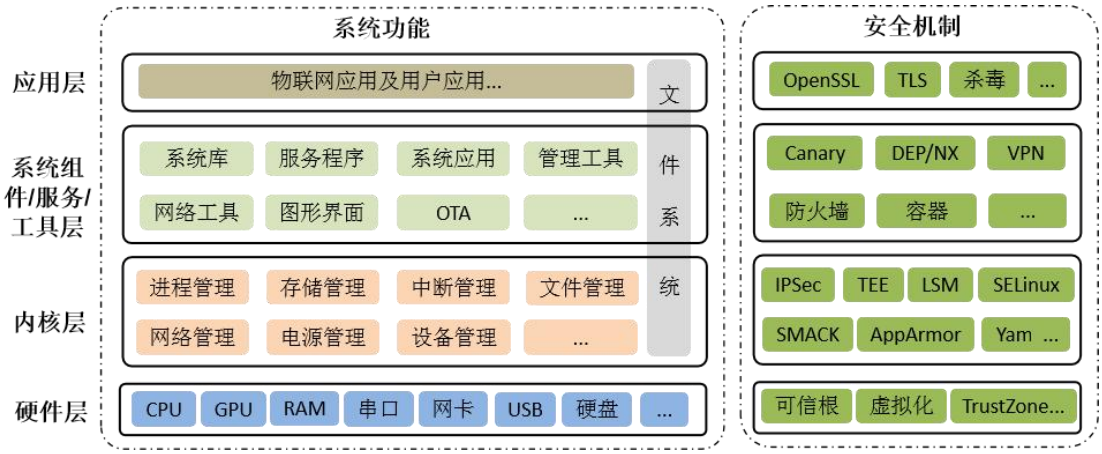

图 3 典型资源丰富型物联网操作系统及安全架构

2. 物联网操作系统安全分析

2.1. 物联网操作系统安全发展态势

1、物联网设备及操作系统面临日趋严峻的安全威胁

2020 年 3 月一种新型 DVR UDP 反射攻击方法被发现[4]。在该次攻击事件中，攻击者采用了一种新的 UDP 反射攻击方法，利用的是某视频监控厂商的设备发现服务 DHDiscover。因为设备厂商的设计不当，当一个外部 IP 地址发送服务发现单播报文时，设备也会对其进行回应，加之设备暴露在互联网上，则可被攻击者用于 DDoS 反射攻击。在设备对 DHDiscover 服务探测报文回应的内容中可以看到关于设备的很多信息，如 MAC（Media Access Control Address）地址、设备类型、设备型号、HTTP Port、设备序列号、设备版本号等。本次攻击流量规模超过 50Gbps。

2020 年 3 月 24 日，ForAllSecure 软件公司的研究员 Guido Vranken 披露了一个基于 Linux 的开源操作系统 OpenWrt 的 RCE 漏洞（漏洞编号为 CVE-2020-7982）。该漏洞存在于 OpenWrt 的 Opkg 包管理器中，由于包解析逻辑中的一个错误，包管理器忽略了嵌入在签名存储库索引中的 SHA-256 检验和，从而绕过了对已下载.ipk 工件的完整性检查。攻击者借助 Opkg 本身的 root 权限以及特制的.ipk 数据包，便可以注入任意恶意代码。值得一提的是，该漏洞在 2017 年 2 月份就被引入代码，距披露时有三年之久。

2020 年 5 月研究人员披露了一个蓝牙协议栈漏洞，攻击者可以利用这个漏洞伪造并欺骗远程配对的蓝牙设备，形成蓝牙冒充攻击（BIAS），其危害性影响数十亿蓝牙设备。蓝牙协议包含了多种身份验证过程，两个蓝牙设备如果要建立加密连接，则必须使用密钥互相配对。但当两个蓝牙设备成功配对之后，下一次它们就能够不经过配对过程而重新连接。 BIAS 攻击就是利用了这个特性。

2020 年 6 月 8 日，安全专家披露了一个名为“Call Stranger” （漏洞编号为 CVE-2020-12695）的新型 UPnP 漏洞，该漏洞影响数十亿台设备，已确认受影响的设备名单包括 Windows PC、Xbox One 以及华硕、贝尔金、博通、思科、戴尔、D-Link、华为、Netgear、三星、TP-Link、中兴等公司的电视和网络设备。该漏洞可能会被远程、未经认证的攻击者滥用，进行反射 DDoS 攻击，并绕过安全系统进行内网渗透及内部端口扫描。

2020 年 6 月 16 日，Treck TCP/IP 协议栈被发现了 19 个 0day 漏洞（又名“Ripple20”）。全球数亿台 IoT 设备，小到家用打印机、摄像头，大到工业控制系统和楼宇自动化设备，都面临被入侵的风险。这一漏洞涉及医疗、航空、运输、

家用设备、企业、能源、电信、零售等行业，众多世界 500 强的公司，如惠普，施耐德电气，英特尔等，都深受其害。

2020 年 7 月 15 日，腾讯安全玄武实验室发布了一项命名为“BadPower”的重大安全问题研究报告，指出市面上现行大量快充终端设备存在安全问题。攻击者可通过改写快充设备的固件来控制充电行为，造成被充电设备元器件烧毁，并可能导致更严重的后果。据保守估计，受“BadPower”影响的终端设备数量可能数以亿计。

上面的物联网操作系统安全事件一方面表明对物联网设备及操作系统而言，协议制订、底层软件实现以及供应链的任意一环出现漏洞，都可能影响数量庞大的物联网设备；另一方面对攻击组织而言，将物联网设备及操作系统相关漏洞利用纳入武器库，利用某一个漏洞即可感染数量相当的僵尸主机，收益极高。

近年来，僵尸网络“推陈出新”，不断改变攻击手法，利用物联网系统服务进行 DDoS 反射攻击。DHDiscover 反射攻击利用了设备厂商的私有协议，物联网设备的发现协议通常基于 UDP 设计，一旦大规模暴露在互联网上，极有可能被用作反射攻击，需要引起重视。另外影响数十亿设备的蓝牙冒充攻击，说明在物联网操作系统设计和实现过程中针对无线通讯的安全不容忽视。

通过绿盟威胁捕获系统检测到近 10 种物联网相关威胁，发现物联网操作系统可被利用的脆弱性涉及弱口令、远程命令执行漏洞等。长期以来，攻击者一直企图采取各种新型手段去探测、攻击并控制物联网设备，不需要花费较高成本即可创建数量庞大的物联网僵尸网络，进而执行传播感染、拒绝服务、域名劫持和钓鱼欺诈等攻击，危害物联网和互联网等重要基础设施和广大普通用户。

2、物联网操作系统安全面临的困境

由于不同的物联网设备的硬件资源、外设等存在较大差异，即物联网设备的碎片化比较严重，导致物联网操作系统也出现同样的情况，即操作系统间的生态互不兼容，操作系统的碎片化愈加严重，可以预见这种情况在未来很长一段时间还将继续存在。对于物联网操作系统的安全功能来说，也存在相同的情况。

以 Arm 的 Mbed OS、阿里巴巴的 AliOS Things、华为的 LiteOS 以及美的智能家电物联网 OS 为例，大多数物联网操作系统都集成了 Arm 公司开发的 MbedTLS。MbedTLS 所占用的最小内存可低至 30KB，但是这对某些物联网设备来说仍然是不可接受的。因此，如何在安全与硬件成本之间进行取舍，是物联网设备厂商和开发者需要共同关注的问题。

3、物联网操作系统安全的研发现状

目前，物联网相关厂商及研发机构主要从系统安全框架构建和安全内核设计着手进行物联网操作系统安全研发。在物联网操作系统安全设计上以整体方案为主，覆盖终端、设备接入、身份认证等。其中物联网终端安全以安全 SDK 为主，

通过将安全 SDK 植入到物联网操作系统中来提高终端与外部通信安全。

市场上的物联网产品在系统架构设计阶段往往忽视安全因素，从而导致物联网产品普遍存在大量的安全漏洞。相关安全企业已逐步提出在物联网设备及操作系统中内置安全模块，为用户提供动态检测、诊断、隔离等安全功能，使得物联网厂商拥有检测物联网设备的安全和可信状态的能力。

内核是物联网操作系统的核心部分，设计安全内核对于构建物联网设备及操作系统安全十分重要。目前，轻量级安全内核的研究主要可分为如下 2 个方向：

(1) 直接改进原有内核的设计增加安全性。例如设计安全内核原型系统，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物联网操作系统。

(2) 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。例如设计独立的、轻量级的可信执行环境，用于保护原有内核的关键操作。

4、物联网操作系统安全相关的法律法规及标准规范发展情况

在安全规范方面，国内外组织近年来不断推进物联网安全标准的制定[5]。在安全体系框架、网络安全、隐私保护、设备安全等方面建立了一系列标准，但整体上侧重于物联网的基础安全框架，应用和服务安全，以及网络与交换安全。针对物联网感控终端安全方面的少量标准也主要侧重于物联网终端设备整体层面，如《20152007-T-469 物联网感知设备安全技术要求》，《GB/T 36951-2018 物联网感知终端应用安全技术要求》，《GB/T 37093-2018 物联网感知层接入通信网的安全要求》等。目前明确针对物联网操作系统相关的标准仅有《GB/T 34976-2017 移动智能终端操作系统安全技术要求和测试评价方法》及《YD/B 173-2017 物联网终端嵌入式操作系统安全技术要求》，尚缺实施指南、检测评估类标准，难以针对大量异构物联网终端设备中多样化物联网操作系统的安全设计和实施进行规范指导。

2.2. 物联网操作系统典型安全问题

物联网操作系统面临的安全问题主要涉及如下方面：

1、非授权访问

由于传统工业控制缺乏对联网场景下的设备访问认证机制，在物联网与工业控制日益融合的情况下，传统工业控制设备的防护机制很容易被恶意用户绕过，造成非授权访问风险；在管理过程中经常使用弱口令，对口令加密保护的强度不够，使得恶意用户可能非法获取系统的控制权限；由于物联网设备应用领域广泛，使得非授权访问恶意攻击造成的影响范围大，进而带来严重的经济损失。

2、数据安全

物联网终端设备硬件容易被攻击者直接获取并通过刷写工具或逆向分析获

取用户的敏感信息；物联网设备及操作系统很容易被非法网络远程入侵，造成用户隐私信息泄露；恶意用户可能通过密码分析破解物联网设备上的加密信息，导致数据泄露；大多物联网操作系统缺乏数据完整性校验和数据加密保护，攻击者通过捕获某些数据包，并重新发给设备，实施非授权行为，给工业生产带来风险。

3、攻击检测及防御

攻击者常常借助病毒、木马、恶意软件等手段，例如使用僵尸病毒，通过自动化脚本组合出物联网终端节点用户名和密码，从而篡改设备配置，使之成为僵尸节点。而随着更多的物联网终端节点被破解，可能形成庞大的僵尸网络，从而破坏物联网系统的可用性。通过部署入侵检测机制可提前阻止对物联网操作系统的恶意攻击，当检测到攻击者有违反安全策略的行为，及时进行异常响应，并采取相应的安全措施。

4、远程升级安全

物联网终端设备由于其部署分散的特点，通常需要对其操作系统进行远程升级，即通过远程发布物联网操作系统升级包完成升级。但在远程升级过程中存在一定的安全隐患，若升级包被篡改，可能导致物联网操作系统在升级过程中被注入恶意代码；若升级包被截获并被逆向分析，可能被恶意攻击者发现可利用的系统漏洞，给系统带来较大的安全隐患。因此在升级过程中需要采取一定的安全机制，保证升级过程的安全，如建立可信通道传输升级包、对升级包加密传输、对升级包进行完整性校验等。

5、通信安全

物联网操作系统有支持多种通信协议的需求，诸如 WiFi、ZigBee 等近距离通信协议，LoRa、NB-IoT 等远距离通信协议，以及基于 MQTT、HTTP 等高层协议。多样的联网方式本身存在一定的安全问题，面临中间人攻击、通信数据被窃取、篡改的风险；恶意用户或进程可能会窃听或破坏终端设备与物联网操作系统的通信，获取系统敏感信息，或是篡改关键通信信息，破坏其通信过程。

6、新技术带来的挑战

新技术融合增大物联网安全风险。随着物联网与人工智能、边缘计算、IPv6、容器、微服务等新技术的加快融合，新技术给物联网带来了功能和性能的提升，但也对现有的物联网安全防护措施带来了新的挑战。

IPv6 将物联网设备暴露于公共网络中，内部和外部系统之间的通信不再有网络隔离，从公共互联网上可以直接访问到物联网内部节点设备，使得物联网设备将更容易遭受网络攻击。

边缘计算从集中式走向分布式部署，并正从通信网络边缘进一步走向物联网应用场景中，形成物联网边缘计算。物联网边缘计算将放大分布式安全风险：一是物联网边缘计算节点数量庞大，复杂性和异构性突出，安全防护策略覆盖困难。

二是物联网边缘计算设备资源和能力有限，难以提供与云数据中心一致的安全能力，边缘节点数据容易被损毁，基础设施软件防护也较为困难。三是物联网边缘计算将采用开放 API 和网络功能虚拟化（Network Function Virtualization，NFV）等技术，开放性的特点容易将物联网边缘节点暴露给外部攻击者。

2.3. 典型物联网场景中的安全风险剖析

2.3.1. 工业控制

随着信息技术（IT）和操作技术（OT）网络数字化转型和融合的加速，物联网(IoT)和工业物联网(IIoT)设备正成为石油和天然气、能源、公用事业、制造业、制药、食品和饮料等行业公司的重要工具。无论是优化单个流程还是整个工厂和其他关键基础设施生态系统，这些设备都有助于提高工业系统的生产效率、生产质量、可靠性以及响应性。

与此同时，效率提升的代价是攻击面的增加，由于更多工业生产环境与互联网直接或间接的接触，导致原本处于独立生态的工控设备将同步处于 IT 类风险之中。

工业物联网场景下的风险主要源于以下几个方面：

1、设备访问凭据问题：大部分工业控制系统在设计之初关注的重点是功能实现，同时默认将其被部署在物理隔离的环境中，并没有过多考虑安全性能，出现了访问设备的用户凭据被硬编码在设备硬件中，或者在管理过程中使用弱口令，保护措施可以被轻易破解或绕过，造成非授权操作等危险后果。在密码保护机制设计实现的时候往往存在非常多的问题，特别是一些国内的工控厂商在设计的时候通常将密码读到上位机组态软件进行对比，这就导致密码保护机制形同虚设。

2、重放攻击：工业控制场景的通信协议设计的关注重点在实时性和功能性，往往欠缺足够的安全设计。由于针对通讯数据缺乏完整性校验功能和足够强度的加密保护，使得攻击者可以捕获设备启动/停止命令的数据包，在不做任何修改的情况下就可以直接发送给控制设备引发设备启动/停止，从而实施非授权操作，给整个业务流程带来风险。

3、拒绝服务攻击：由于开发人员安全开发能力缺乏和安全意识不足，导致设备功能在设计和实现时不具备过多的容错功能，即缺乏针对畸形报文的异常处理能力，使得攻击者可以利用异常数据和非常规操作对工控设备进行拒绝服务攻击。而工控系统中一个节点设备的异常就有可能导致整个生产线的瘫痪。

4、供应链攻击：供应链攻击是一种面向物联网开发人员和厂商的新兴威胁。攻击方法是通过在合法应用、服务、设备中植入恶意代码，基于预先设定的触发

条件完成针对目标业务系统的攻击。供应链可划分为开发、交付、运维三个大的环节，每个环节都可能会引入供应链安全风险从而遭受攻击，而且上游环节的安全问题会传递到下游环节并被放大。值得注意的是在供应链攻击中受到攻击的是上游厂商，受到威胁的则是上下游厂商。基于供应链攻击的常用方法：

(1) 利用供应商的产品植入恶意软硬件模块。
(2) 利用第三方组件（打包、伪装、代码植入、硬件植入）。
(3) 利用开源代码库中包含的漏洞。
(4) 利用“内鬼”在源码中植入恶意功能。
(5) 恶意接管（利用社区项目管理职权注入恶意代码）。
(6) 利用非官方售后服务（安装恶意软件、植入恶意硬件）。
(7) 工业设备生产厂商预留的运维后门。

当前工业互联网设备正处于快速增长的发展阶段，设备制造商往往只注重产品的可用性和易用性，受限于硬件资源很难实现细粒度的系统安全措施。同时，真实的制造环境中往往需要多个厂商、多种类型的工业互联网设备协同工作，在缺乏统一安全技术要求规范来保证整个系统交互安全的情况下，大大增加了攻击面，给工控系统的安全建设带来严峻的挑战。

2.3.2. 智能家居

智能家电中的物联网操作系统的一个重要作用是实现可信、安全的连接，包括智能家电与控制端应用程序之间的连接，智能家电与远程云服务器之间的连接，智能家电与智能中控平台的连接、以及智能家电相互之间的连接。智能家电的安全运行需要物联网操作系统支持多种通信协议，典型的如 WiFi，BLE，ZigBee，NB-IoT，4G/5G 等，并且能保证多种协议同时工作时的安全可靠。依靠物联网操作系统，用户可以让智能家电完成设备的配网连接、注册绑定、远程控制、状态推送、升级更新等功能。在智能家电的工作过程中，其上搭载的物联网操作系统应提供对通信数据的机密性和完整性的保护以及对家电设备身份的安全认证，防止恶意攻击者非法控制家电设备或伪造家电设备控制指令等，从而保障智能家电使用过程的安全。

智能家电需要物联网操作系统解决的网络安全威胁主要包括以下几种：

1、系统安全：恶意攻击者通过仿冒升级服务器方式向智能家电发送异常升级包，修改操作系统代码，破坏其完整性，实现对操作系统功能和数据的滥用和破坏。

2、非授权访问：非授权用户绕过家电操作系统中设备配网过程和注册过程的身份鉴别机制，向家电设备发送非授权指令，获取敏感数据和用户数据，或对敏感数据和用户数据进行恶意操作，或滥用操作系统的安全功能。

3、数据安全：恶意用户通过密码分析等手段访问操作系统存储在存储器件

上的数据，造成数据泄露。或者在家电设备未上电，操作系统未运行的情形下，恶意用户通过对 IoT 设备实施物理攻击，直接拷贝或篡改存储设备上所有数据，造成数据泄露或损坏。

4、网络安全：恶意用户或进程通过密码分析等手段侦听或破坏智能家电操作系统与外部 IT 实体之间的通信，获取敏感信息、破坏数据保密性；或者攻击者可能介入智能家电操作系统与远程实体的通信，改变智能家电设备和其他端点之间的通信。

2.3.3. 智能表计

能源关系着国计民生，能源计量作为能源行业重要一环，促使能源计量的数字化升级发展迅速。物联网技术驱动的智慧水务、智慧燃气也已实现规模化应用，据行业调研数据，2020 年中国物联网燃气表出货量突破千万台，年增长率达 30%以上。

但目前智能表计行业尚属于发展初期，除了少数大型企业，“重发展而轻安全”是行业普遍现象，这将为能源行业数字化转型升级留下很大的安全隐患。国外的几个相关安全事件值得借鉴：2014 年西班牙智能电表被曝出安全漏洞，可被利用实施电费欺诈，甚至控制电路系统导致大面积停电；2021 年施耐德智能电表曝严重漏洞，可被远程强制重启等。

物联网智能表主要包括电表、水表、燃气表、热力表等。作为典型的物联网感知终端，物联网智能表具有低功耗（电表不要求低功耗）、低带宽、资源受限等特点，无法应用复杂的安全防护手段。当前物联网智能表主要面临以下安全风险：

1、物理安全：很多智能表设备为户外安装且无人看护，攻击者容易接触到终端硬件，可以利用工具直接从硬件中提取固件和敏感数据，进一步通过逆向分析寻找漏洞或提取密钥等敏感信息。

2、系统安全：固件存在漏洞可能被攻击者恶意利用，固件更新过程中可能遭到篡改甚至替换；目前相当比例的物联网抄表系统仅靠设备或服务端的唯一标识进行身份鉴别且未实施双向认证，终端和云端身份容易被仿冒。物联网智能表设备被非法控制可能导致大规模停水停电，影响公共安全。

3、数据安全：物联网智能表涉及的水、电、气、暖等基础设施的数据属于社会敏感数据，通常包含用户隐私信息，大规模的基础设施数据泄露可能影响数百万人的生活甚至国家安全。

4、通信安全：当前智能表联网方式多种多样，有 NB-IoT、LoRa 等低功耗远距离通信方式，也有采用网关+近距离无线通信的方式，其中网关+近距离无线通信的实现又有多种方案。多样化的联网方式存在诸多安全问题，面临中间人攻击，通信数据存在被窃取、篡改的风险，可能造成企业用户的信息泄露和财产

损失。

2.3.4. 车联网

随着汽车智能化、网联化、共享化程度的不断提升，面临的信息安全威胁也越来越多样性，从通信窃听、OTA 数据包篡改、钥匙重放攻击到现在针对语音控制的“海豚音”攻击、针对自动驾驶“道路识别”、“自动雨刷”的对抗样本攻击，车联网的信息安全形势越发严峻。车联网操作系统作为网联汽车的核心，向上承接各种业务、通信等应用功能，向下承接底层资源调用和管理，是车联网安全的基石。早在 2016 年，有安全研究员利用漏洞对特斯拉进行了无物理接触远程攻击，实现对特斯拉驻车和行驶状态的远程控制。其中一个关键环节就是通过 Arm Linux 漏洞 CVE-2013-6282 获得了 CID 的 root 权限，进而以 CID 为跳板进一步渗透进入 IC、Parrot 和 Gateway，从而打通了整个攻击链条。

当前，车联网的安全风险主要包含硬件安全风险、固件安全风险、操作系统安全风险、数据安全风险、远程升级安全风险等部分。

1、硬件安全：硬件安全是车联网安全的最基本要求。传统 IT 的核心资产服务器都放在攻击者无法物理接触的密闭场所，而汽车却截然不同。攻击者在信息搜集的阶段往往通过印制主板上的丝印信息来获取攻击对象的具体信息，甚至可能通过调试接口直接非法访问系统、提取系统中的信息。

2、固件安全：固件通常被存储在外部存储器中，如果攻击者提取出固件后对其进行篡改，再刷写回去，将对汽车造成巨大的安全隐患。其次，固件中往往包含了系统或应用的许多信息，通过固件分析扫描，攻击者还可以直接检测出系统存在的 CVE 漏洞、不安全配置、甚至是明文的密钥。

3、系统安全：操作系统控制了所有应用对硬件、软件资源的访问，如果存在严重的安全风险相较于应用而言，将是毁灭性的。常见的风险包括内核或关键部件存在已知漏洞、网络防护策略配置不当、敏感信息泄露、安全行驶参数配置错误、更新策略不完善和非安全启动。

4、数据安全：当前，国内外越发重视数据安全、个人隐私安全，汽车为人类提供出行服务的同时，产生、处理、存储了大量重要数据。舒适化的发展离不开司机、驾驶人员的个人信息、驾驶数据、行为习惯，智能化的发展离不了道路、路侧设备、道路其他使用者、行人的数据。这些数据如果被非法利用可能会威胁整个社会的安全。

5、远程升级安全：当前，OTA 技术已经非常成熟，但升级过程的安全性依旧存在许多安全风险。2020 年，联合国批准了 R156 法规，对汽车软件升级提出了具体的要求。如何保障升级流程安全、升级包传输安全、ECU 升级安全依旧充满挑战。

2.3.5. 视频网

公共安全视频监控建设联网应用，是新形势下维护国家安全和社会稳定、预防和打击暴力恐怖犯罪的重要手段，是动态化、信息化条件下完善社会治安防控体系、深化平安中国建设的重要基础性工程，对于提升城乡管理水平、创新社会治理体制具有重要意义。

公安视频网场景，海量摄像头等物联网终端均采用 Linux 各剪裁版本的操作系统，这些版本的物联网操作系统因为裁剪程度不同，引入的开源组建情况不同，安全等级不同。

1、摄像头系统脆弱性问题

由于摄像头终端自身系统的脆弱性，较多的安全漏洞，导致其很容易被入侵，劫持为“肉鸡”发起 DDOS 攻击等危害公共安全行为，包括前端摄像头弱口令、系统漏洞等安全状态无法实时监测都对用户网络安全造成极大威胁。

2、摄像头非法接入问题

以公安场景为例，海量摄像头部署在城市各个角度，边界非常广，终端非法接入网络，或者非法将摄像头换成 PC 设备的行为很容易，一旦发生即可与视频网核心服务进行网络可达，很容易造成内部网络攻击行为或者病毒传播行为。

3、摄像头等物联网终端的固件安全问题

在公安视频网场景，这些摄像头一旦感染病毒，就会横向扩散，产生类似与 2016 年 MIRAI 病毒的攻击危害。

4、摄像头数据安全问题

这些年摄像头因为对外开放的 WEB 服务权限过大导致的数据泄漏事件屡见不鲜，摄像头在城市各个角度部署，采集大量涉及民生的车辆、人脸图像、视频数据，一旦泄漏就可能造成恶劣的社会影响、甚至是经济损失。

总体来说在视频网场景，亟需建立统一的标准的物联网安全操作系统，而不是各自厂家独立裁剪开源操作系统，来规避因为开源组件、操作系统漏洞造成的安全风险。

3. 物联网操作系统关键安全技术

基于典型物联网场景中的安全风险分析，设备安全、通信安全、系统安全和数据安全是物联网中的几大主要安全风险。在这当中，设备的可信接入，授权访问，通信的可靠和安全、系统的健壮性和可追溯性、数据的保护和安全是物联网场景中最为关注的安全问题。为更好的应对这些安全问题，可以在物联网操作系统研发过程中选择利用下述关键安全技术，如表 1：

表 1 物联网操作系统关键安全技术

安全威胁	安全问题	关键安全技术
非授权用户登录越权攻击、非法设备接入	维测数据、版本文件、维测程序遭到破坏，无法进行维测、版本无法升级	身份鉴别技术 访问控制技术
盗窃、破解机密信息	核心数据、隐私泄露	密码技术
利用网络漏洞进行攻击	设备连接故障，关键信息被侦听或获取	网络连接及网络通信安全技术
利用欺骗伪造攻击	维测数据、版本文件、维测程序被非法篡改	可信及 TEE 安全技术
所有类型攻击	无法对攻击行为进行追溯以及及时处置	日志审计及安全态势感知技术
利用非法版本和程序进行攻击	系统被非法控制，成为肉鸡	系统升级安全技术
利用系统漏洞进行攻击	系统崩溃	资源竞争安全技术

下面的子章节将对这些关键安全技术进行详细说明。

3.1. 身份鉴别技术

身份鉴别是证实主体的真实身份与其所声称的身份是否相符的过程，也是正确实施访问控制机制的前提。物联网应用系统和网络接入都需要依赖身份标识和身份认证，确保正确的设备接入正确的网络，传输正确的数据，执行正确的动作。

物联网设备应用场景中涉及三个身份鉴别环节：设备身份鉴别、远程通信实体身份认证和用户身份鉴别。

1、设备身份鉴别

物联网设备通常需要与管理平台或用户账户进行绑定，绑定过程需要对设备进行身份鉴别。物联网设备通常使用设备唯一标识符作为其身份的证明，例如通信模块的 IMEI（International Mobile Equipment Identity）号、MAC（Media Access Control Address）地址等。

为了支持身份鉴别，物联网操作系统通常会提供访问设备唯一标识符的功能或接口，并通过访问控制机制，防止其被非法篡改。常见技术存在以下两种：

(1) 采用用户授权机制，在应用代码访问设备唯一标识符时，通知用户并请求用户授权；

(2) 采用密码学技术，保护设备唯一标识符的完整性。

2、远程通信实体身份认证

为保护物联网设备上敏感数据的安全，防止设备被非法远程操控，物联网操作系统需支持对远程通信实体的身份认证。常见身份认证技术包含以下几种：

(1) 基于对称密码算法的身份认证方式，即使用共享密钥或其生成的密钥对通信双方的身份认证信息进行加解密。为保证安全，物联网操作系统需将所用对称密钥妥善保存在设备端，避免被非授权读取或被篡改。

(2) 基于非对称密码算法的身份认证方式，即通过“公钥交换”方式实现实体身份认证。物联网设备需先获得远程通信实体的公钥，远程通信实体使用自己的私钥对身份信息或通信数据进行签名，物联网操作系统使用远程通信实体对应的公钥对该签名进行验签，从而验证远程通信实体的身份。进一步的，可使用类似 X.509 的证书体系，通过证书链验证远程通信实体身份。

(3) 基于 SM9 标识密码算法的身份认证方式，既使用国家密码管理局于 2016 年 3 月 28 日发布的 GMT 0044-2016 SM9 标识密码算法对通信双方的身份进行鉴别。SM9 标识密码算法是一种基于双线性对的标识密码算法，可以运用用户的身份标识生成公私钥对，用于数字签名、数据加密、密钥交换以及身份认证等。SM9 密码算法的应用与管理不依赖于数字证书、证书库或密码库，使得物联网身份认证等安全应用更易于部署和使用。

3、用户身份鉴别

若物联网设备具备用户体系，则物联网操作系统应支持用户身份鉴别机制。目前，物联网场景下最常见的设备端用户身份鉴别方式为口令验证和生物信息识别。

对于口令验证方式，用户需输入用户身份标识及一串他人无法知晓的秘密信息来表明自己的身份，物联网操作系统验证用户输入的信息与其已存储的信息是否一致，若一致则通过验证。用户身份标识是用户在物联网操作系统中注册的账户名，秘密信息即为用户设置的口令，可为数字、字母、特殊符号等内容的组合。物联网操作系统一般会根据应用场景安全要求，采用合适的口令长度和复杂度要求，例如多数物联网设备会要求用户口令不得少于 6 个数字。为了防止口令在输入过程被泄露，物联网操作系统还会在口令输入时提供受保护的反馈显示，例如仅向用户提供非鉴权数据（如圆点、星号等）作为鉴别数据输入的反馈，鉴别失败时仅将最少的反馈（如输入的字符数）提供给鉴别的用户等。此外，对于用户口令在设备上的存储，物联网操作系统一般会采用只存储口令哈希值或密文的方

法进行保护，避免口令明文数据被泄露。口令是一种简单易行的身份鉴别手段，但是因为容易被猜测而比较脆弱，易被非法用户利用。

对于生物信息识别方式，用户通过物联网设备上的生物信息采集模块录入生物信息并生成特征模板，在验证时，物联网设备会比对当前采集的用户生物信息与存储的特征模板的一致性，从而给出身份鉴别结果。在实际应用时，物联网操作系统需提供生物信息采集、存储、比对等环节的安全保护。目前已有部分物联网操作系统采用可信执行环境等技术，来保护生物特征信息比对过程不被干扰，并通过访问控制、加密存储等机制保护存储在设备上的生物特征信息不被篡改。

此外，不少增强级物联网操作系统加强了鉴别力度，采用了多因素鉴别机制，在每次用户登录系统时，采用受安全管理中心控制的口令、令牌、基于生物特征、数字证书以及其他具有相应安全强度的两种或两种以上的组合机制进行用户身份鉴别，并对鉴别数据进行机密性和完整性保护。

3.2. 访问控制技术

物联网设备上存在多类敏感资源，例如用户数据、传感器数据、设备接口等，为了提供资源的安全性保护，物联网操作系统需设计相应的访问控制机制，以提供资源的受控访问。

访问控制一般分为自主访问控制（Discretionary Access Control，DAC）和强制访问控制（Mandatory Access Control，MAC）两种形式。

自主访问控制（DAC） 是最常用的一类访问控制机制，是用来决定一个用户是否有权访问客体的一种访问约束机制。自主访问控制机制是用于保护信息系统的资源不被非法用户访问的一种有效手段，但它有一个明显的缺点，就是这种控制是自主的，而用户不是安全专家，很容易被善于伪装的攻击者所欺骗，从而给恶意代码渗透留下了机会。所以，系统需要采取更强的访问控制手段，如强制访问控制机制。在强制访问控制（MAC）中，系统中的每个任务、文件、通信机制等都被赋予了相应的安全属性。并且这些安全属性是不能改变的。

强制访问控制（MAC）可以弥补自主访问控制（DAC）在权限控制过于分散、防范木马型攻击等方面不足，但也存在过度强调保密性，管理不够灵活，适用范围比较小、易用性比较差的缺点，通常与自主访问控制（DAC）结合使用，主体只有通过了自主访问控制（DAC）和强制访问控制（MAC）的检查后，才能访问客体。

此外，业界还有基于对象的访问控制（Object-based Access Control, OBAC）、基于任务的访问控制（Task-based Access Control, TBAC）、基于角色的访问控制（Role-based Access Control，RBAC）和基于属性的访问控制（Attribute-based Access Control，ABAC）等机制。

基于对象的访问控制（OBAC）是将访问控制列表与受控对象或受控对象的属性相关联，将访问控制选项设计为用户、组或角色及其对应权限的集合。OBAC 使得当受控对象的属性发生改变时，无须更新访问主体的权限，只须修改受控对象的相应访问控制项即可，从而减少了访问主体的权限管理，降低了授权数据管理的复杂性。

基于任务的访问控制 （TBAC），以面向任务的观点，从任务（活动）的角度来建立安全模型和实现安全机制，在任务处理的过程中提供动态实时的安全管理。一个工作流的业务流程由多个任务构成，而一个任务对应于一个授权结构体，每个授权结构体由特定的授权步组成，一个授权步的处理可以决定后续授权步对处理对象的操作许可。

基于角色的访问控制（RBAC）是将访问许可权分配给一定的角色，用户通过饰演不同的角色来获得角色所拥有的访问许可权。RBAC 通过给用户分配合适的角色，让用户与访问权限相联系，角色成为了访问控制中访问主体和受控对象之间的一座桥梁。

基于属性的访问控制（ABAC）将主体和客体关联的属性作为授权决策的基础，利用属性表达式描述访问策略。ABAC 能够根据相关实体属性的变化，适时更新访问控制决策，从而提供一种更细粒度的、更加灵活的访问控制方法。

3.3. 密码技术

密码技术是物联网安全中经常被使用到的技术，在提供设备信任根、安全启动、身份识别、安全通信、保护数据安全等多种场景都有广泛使用。物联网系统中经常使用的密码学算法可分为三大类，对称密钥算法、非对称密钥算以及法哈希算法。

1、对称密钥算法

对称密钥算法即加密和解密过程使用相同密钥的加解密算法。常见的对称密钥算法包括 DES，TDES，AES，SM4 等，它们都是分组密码算法（块加密算法），能够对固定长度的明文数据块进行加密。分组密码算法还可以结合各种操作模式对超过块长度的明文数据进行加密，例如 ECB，CBC，OFB，CFB，CTR 等，根据各种不同模式的特点可以用于不同的场景；还有一些操作模式，例如 CCM、GCM，在对数据进行加密的同时还能够生成额外的校验数据块，在保护数据机密性的基础上增加数据完整性和消息来源合法性的保护。

对称密钥算法的优点是运算量相对较小，效率高，常用于对大数据量的消息或数据流进行加密；不足是加解密双方使用相同的密钥，密钥的安全存储、管理、分发等成为一个挑战。

2、非对称密钥算法

非对称密钥算法使用一组成对的私钥和公钥进行消息处理。公钥顾名思义是的公开的，没有机密性保护需求，只有私钥是机密的。非对称密钥算法在物联网中通常作为密钥交换协议和数字签名安全扩展协议的一部分，用于确保机密性、身份验证或不可抵赖性。

非对称密钥算法使用的场景包括密钥交换、数字签名和数据加密，由于公钥算法基于大数运算的数学难题，其运算相对复杂且运算量大，效率没有对称密钥算法高，所以往往不会用于大量数据的加解密操作，更多的使用场景还是密钥交换和数字签名。

3、哈希算法

也称为散列算法、杂凑算法。哈希算法是一个单向映射，其特点是输入数据的任何一个微小变化都会带来摘要数据的巨大变化。这种单向转换函数在验证数据是否遭到篡改（即消息完整性）方面扮演重要角色。哈希算法还可以结合对称密钥，生成消验证码（Message Authentication Code，MAC）、基于哈希的消息验证码（HMAC）或用于密钥派生（KDF）。常见的哈希算法包括 MD5、SHA1、SHA2、SHA3、SM3 等。

密码学算法本身的安全性和强度与算法使用的密钥长度及相关参数有关，某些旧的算法或短密钥长度已经无法满足当下及未来的安全需求。在追求更高安全性的趋势下，密码算法和建议的密钥长度在不断发展。根据 GlobalPlatform 在 2021 年的 Cryptographic Algorithm Recommendation 中建议[6]，新产品应当使用至少达到相当于 128 位安全强度的算法，因此 DES/3DES 算法，ECB 和 CTS 模式 MD5、SHA-1、SHA-224 以及低于 3072 位的 RSA 等算法已经不推荐在新产品中使用。TLS v1.3 在很大程度上也弃用了 v1.2 版本的很多密码学套件，转而使用更强大的算法，包括基于 HMAC 提取和扩展密钥派生函数（HKDF），以及带有关联数据认证的加密（Authenticated Encryption with Associated Data，AEAD），确保满足数据机密性、完整性和真实性的需求。再比如 CNSA Suite 也取代了早期的 NSA Suite B，并建议使用更稳健的参数来保护高机密级别的信息。

物联网操作系统通常包含相应的密码算法模块来支持密码算法操作。一种方式是纯软件实现，它的好处是可以支持各种不同的算法以及不同的硬件平台。另一种方式是结合底层芯片硬件的加解密单元，通过硬件实现各种算法。例如比较为大家所熟悉的 mbed TLS 可以通过 Alternate 函数的方式，利用芯片硬件加解密模块和随机数发生器完成如 AES、RSA、SHA、ECC 等算法操作和随机数生成。嵌入式设备上支持加解密算法可能有多方面因素需要考量，例如 CPU 负载、算法性能、Flash 和 RAM 占用空间、对功耗的影响等，在资源受限的系统中，通过芯片硬件加解密模块完成密码学操作在这些方面都有比较明显的优势。部分芯片的硬件加解密引擎（比如 STM32U5 的 PKA，SAES）还具备防侧信道攻击、错误注入攻击的能力，能够给密码算法的执行带来更进一步的安全保护。

此外还有一些新型加密技术也值得关注，例如可搜索加密算法、安全多方计算算法、同态加密算法、零知识证明、量子密码算法以及轻量级密码技术。其中轻量级密码技术通常会综合考虑存储需求、门面积、延迟、吞吐量、能耗以及硬件实现效率等因素，因而更加适合部署于资源及性能受限的物联网终端（例如收录于 ISO29192 国际标准的 CLEFIA 和 PRESENT 轻量级分组密码算法等）。

3.4. 物联网通信安全技术

物联网通信技术能够使物联网将感知到的信息在不同的终端之间进行高效传输和交换，实现信息资源的互通和共享，是物联网各种应用功能的关键支撑。在物联网应用中，通信技术包括 Wi-Fi、Bluetooth、ZigBee、NFC、NB-IoT、LTE、5G、LoRa、USB、RS485、Ethernet、CAN 等，网络传输层包括 IPv4、IPv6、6LoWPAN、TCP、UDP 等通信协议，应用层包括等 MQTT、CoAP、DDS、XMPP、AMQP、HTTP 等协议，协议分层如图 4 所示。

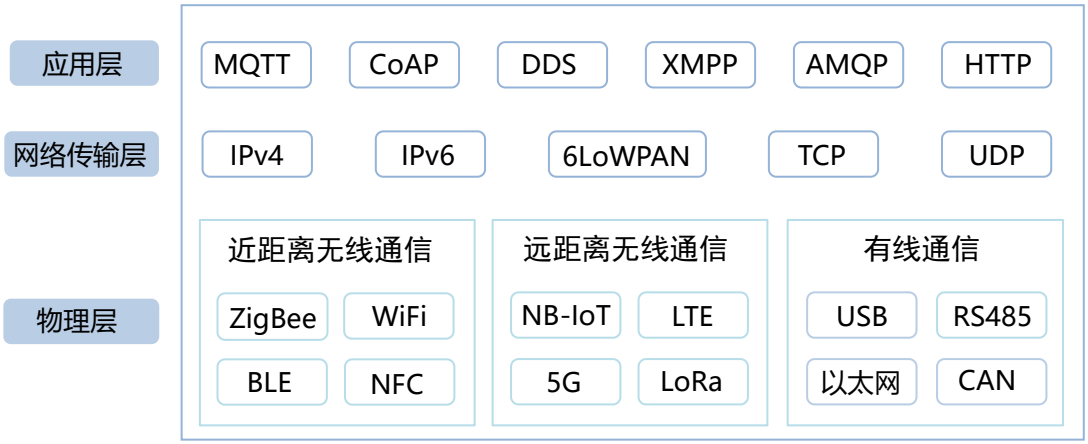

图 4 常见物联网通信协议

物联网通信安全技术主要包括：

1、认证机制：在通信前物联网设备必须对所有合法用户进行身份验证（开放式认证和共享秘钥认证），防止非认证用户非法接入。实现这种认证的方法包括静态口令、双因素身份认证、生物认证和数字证书等。

2、加密机制：加密主要用于防止对敏感数据和物联网设备的未经授权访问，加密算法主要有对称加密、非对称加密和摘要算法等。

3、密钥管理：密钥是整个系统安全的基石，是网络安全和通信保护的关键点，密钥管理包括密钥的生成、分发、更新等。

物联网常用无线连接技术安全机制如表 2 所示：

表 2 物联网常用无线连接技术安全机制

类别	密钥管理	数据加密	认证与完整性保护
WiFi	预置共享密钥	AES-CCM	CBC-MAC
ZigBee	预置链接密钥，信任中心生成/更新网络密钥	AES-CCM	CBC-MAC
BLE	Passkey， ECDH	AES-CCM	CBC-MAC
LoRa	预置 NwkSKey 及 AppSKey	AES128	CMAC
NB-IoT	基于 SIM 的预置共享密钥，多级密钥衍生	SNOW 3G/AES/ZUC	基于 Milenage 算法的 AKA 鉴权，双向认证
LTE Cat.1	基于 SIM 的预置共享密钥，多级密钥衍生	SNOW 3G/AES/ZUC	基于 Milenage 算法的 AKA 鉴权，双向认证

网络 IP 化是物联网通信技术的趋势，资源丰富型物联网设备一般支持完整的 TCP/IP 协议栈，而在一些带宽资源极度受限或功耗要求严格的通信环境下，如 ZigBee(802.15.4)、BLE(802.15.1)等，可采用一种非常紧凑、高效的 IP 实现 6LoWPAN——基于 IPv6 的低速无线个域网标准。网络传输层采用的常见安全技术主要有 IPSec、TLS、 DTLS 等。

IPSec 是一个协议簇，通过对 IP 协议的分组进行加密和认证来保护 IP 协议的网络传输协议簇。IPSec 可以实现以下 4 项功能：①数据机密性：IPSec 发送方将包加密后再通过网络发送；② 数据完整性：IPSec 可以验证 IPSec 发送方发送的包，以确保数据传输时没有被改变；③数据认证：IPSec 接受方能够鉴别 IPSec 包的发送起源，此服务依赖数据的完整性；④防重放:IPSec 接受方能检查并拒绝重放包。

IPSec 主要由以下协议组成：

一、认证头（AH），为 IP 数据报提供无连接数据完整性、消息认证以及防重放攻击保护；

二、封装安全载荷（ESP），提供机密性、数据源认证、无连接完整性、防重放和有限的传输流（traffic-flow）机密性；

三、安全关联（SA），提供算法和数据包，提供 AH、ESP 操作所需的参数。

四、密钥协议（IKE），提供对称密码的钥匙的生存和交换。

TLS 全称是 Transport Layer Security，TLS 是建立在传输层 TCP 协议之上的协议，服务于应用层，它的前身是 SSL（Secure Socket Layer，安全套接字层），它实现了将应用层的报文进行加密后再交由 TCP 进行传输的功能。TLS 协议主要解决以下三个网络安全问题：①保密性，通过加密实现，所有信息都加密传输，第三方无法嗅探；②完整性，通过 MAC 校验机制，一旦被篡改，通信双方会立

刻发现；③认证，双方都可以配备证书，防止身份被冒充。

TLS 协议由两层组成：握手协议和记录协议。握手协议用于客户端和服务端在加密通信之前进行算法套件和加密密钥的协商；记录协议为 TLS 上层子协议提供分片、消息加密及加密后报传输，同时对接收到的数据进行验证、解密、重新组装等功能。

DTLS 全称是 Datagram Transport Layer Security，即数据包传输层安全性协议。DTLS 基于 UDP 场景下数据包可能丢失或重新排序的现实情况下，为 UDP 定制和改进的 TLS 协议。与 TLS 协议一样，DTLS 同样由两层组成: 记录协议和握手协议。DTLS 记录协议在报文段头部增加一个显式的序列号，缓存乱序到达的报文段和重传机制实现了可靠传送。而握手协议 DTLS 与 TLS 相比有 3 个主要的变化：①添加了一个无状态的 cookie 交换来预防 DOS 攻击；②修改了握手报文头来处理消息丢失、重组和 DTLS 消息分片（从而避免 IP 分片）；③重发计时器用于处理消息丢失。

安全通信协议实现方面，资源丰富型物联网设备可选择 OpenSSL，而对于资源受限的物联网设备可以选择 Mbed TLS、wolfSSL、NetX Secure、OneTLS 等轻量级的嵌入式安全通信协议栈。

3.5. 可信计算及可信执行环境技术

1、可信计算技术

可信计算是从计算机的芯片、硬件结构、操作系统等方面采取综合措施，通过建立一种特定的可信度量和验证机制，使计算平台自身具备安全防护能力的一个技术体系。国际上，可信计算组织（Trusted Computing Group，TCG）和国际标准化组织（ISO）发布了一系列针对可信计算的技术规范，定义了体系结构、功能和接口，如可信平台模块（TPM）相关规范、TPM 软件栈相关规范、PC 客户端相关规范等；并在谷歌、微软、思科等公司的产品和服务中得到了广泛的使用。在国内，也相继研制了《可信计算平台密码方案》和《可信计算平台密码技术规范》等规范；国家“网络安全法”中要求“推广安全可信的网络产品和服务”，等保 2.0 标准从安全保护能力第一级至第四级均提出可信验证的要求，在第三级、第四级更是提出动态可信验证的要求，同时针对物联网提出专门的扩展要求；工信部提出积极探索可信计算等网络安全的新理念、新架构，促进网络安全理念和技术的创新。国内已经形成了较为完备的可信计算产业；国家关键基础设施、重要企业等也结合自身领域特点，积极设立在领域内使用可信计算技术的规范、积极采用可信计算技术对业务系统进行防护，提高关键业务的安全性。

可信计算技术的基本思想是在物联网设备系统中建立一个可信根，该可信根为系统信任的起点。物联网设备系统从可信根开始，对硬件平台、操作系统、应

用程序及重要配置参数等逐级进行度量、验证，并采取防护措施确保物联网设备的数据完整性和行为的预期性，将信任扩展到整个物联网设备及系统，从而提高物联网设备的可信性。在此基础上，利用可信的远程证明、可信网络连接等技术，对多个物联网节点的可信性、节点间网络连接等进行统一安全可信管理，最终形成可信的物联网系统和服务。

根据物联网系统安全性高、系统多设备组成复杂、功能通常较为单一的特点，可利用可信计算技术实现对物联网设备从最底层的物理硬件、BIOS/BootLoader 等固件、操作系统、业务应用及重要数据等全部执行环节进行防护，如下：

(1) 具备形成整体可信链的能力：可信计算的防护具有基础性、全执行周期的特点，建议物联网设备在物理层面支持可信计算技术，在物联网操作系统加载前对物理硬件、固件/操作系统加载器等执行组件进行可信验证，物联网操作系统应具备与相应可信计算整体功能配合的能力，形成完整的可信链。

(2) 具备可信验证功能：物联网操作系统应提供针对应用程序、重要配置参数等的可信验证功能。针对不同的物联网设备和物联网业务特点，可信验证应包括应用程序静态可信验证和动态可信验证等能力。当可信验证结果为不可信时，应支持阻断执行等操作，防止系统、服务遭到异常篡改。

(3) 具备接受可信安全管理中心统一可信管理的功能：物联网操作系统应支持接受可信安全管理中心统一可信管理的功能，如可信策略管理、可信状态上报、可信日志上报等功能。

(4) 具备可信报警功能：物联网操作系统应提供可信报警功能。当物联网设备上的业务应用软件、重要配置参数等发生可信验证结果为不可信时，支持报警、并向可信安全管理中心上报。

2、DICE

为了增强物联网设备、系统和应用的安全性，可信计算组织（TCG）组织建立了设备标识组合引擎技术（Device Identifier Composition Engine，DICE）工作组。DICE 工作组旨在探索适用于有或没有 TPM 的系统和组件的新的安全和隐私技术。目标是开发新的方法，以最小的硅需求增强安全性和隐私性。

DICE 基于简单的硬件要求，可以适用于大多数系统或组件，提供基于硬件的身份和认证、密封、数据完整性、设备恢复和更新的能力。在 DICE 架构中设备启动是分层的。从只有制造商和 DICE 知道的唯一设备秘密（Unique Device Secret，UDS）开始，创建每一层特有的秘密和硬件配置。这种派生方法确保如果在特定设备上引导不同的代码或配置，则设备上的秘密将是不同的。每个软件层都保证它收到的秘密只有它自己知道。如果存在漏洞而且一个秘密被泄露，修补代码自动创建一个新的秘密，可以有效地重新设置设备 Layer0 以上其他层的密钥。UDS 受到硬件访问保护，仅有 DICE 对其可读。而且 DICE 保存在只读存储上，不能被篡改。每一层通过加密单向函数为下一层计算一个秘密。每一层都

保护着它所接收到的秘密。在每层软件执行后下层软件执行前，除了交由下层的处理信息，其余在寄存器和内存上的信息都需被全部抹除。如图 5 所示：

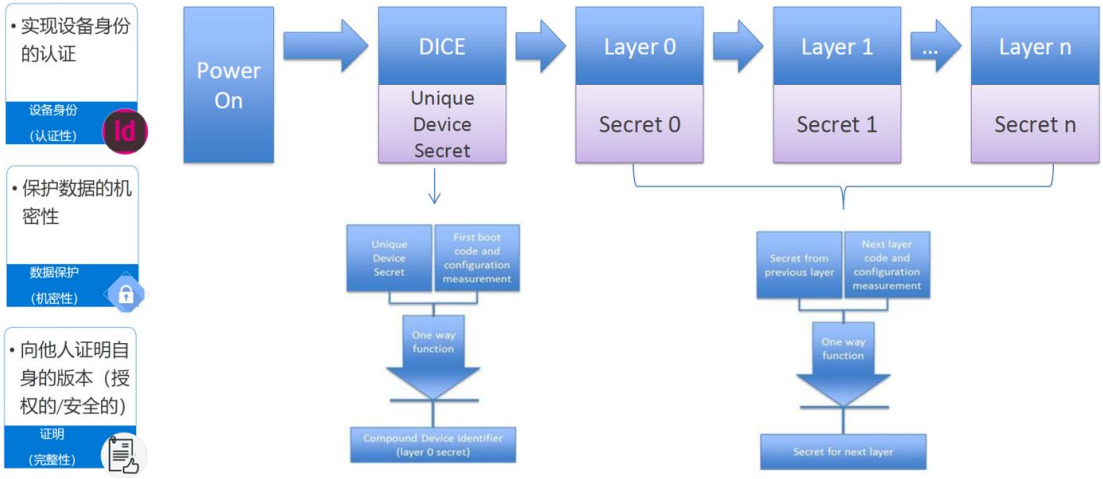

图 5 DICE-设备标识组合引擎

3、可信执行环境技术

物联网操作系统和应用的漏洞会导致大规模物联网设备被恶意代码注入，从而导致用户隐私数据泄露。目前应用于物联网系统和应用层面的数据隔离和虚拟化技术不足以提供可靠的安全隔离保护，攻击者仍然可以通过恶意代码越界访问整个设备硬件中存储的数据，而可信执行环境技术可以降低上述安全威胁。

可信执行环境 (Trusted Execution Environment ,TEE)是 CPU 芯片层面上单独划分出来的一片“区域”，是独立于开放系统（例 FreeRTOS、mbed OS、OneOS、AliOS 等等系统）而存在的可信的、隔离的、独立的执行环境，与开放系统并行运行。TEE 可给数据和代码的执行提供一个更安全的空间，并保证它们的机密性和完整性，对安全性要求更高的业务可部署到 TEE 中。

TEE 依赖于硬件隔离技术，将运行环境分为安全世界和非安全世界。开放系统运行在非安全世界，可信执行环境运行在安全世界。这两个世界完全是硬件隔离的，非安全世界运行的系统和应用程序只能访问非安全世界的内存和资源，安全世界运行的系统和应用程序可以访问安全和非安全世界的内存和资源。即使开放系统被破坏，处于 TEE 隔离的环境中的数据和代码也会受到保护。

目前各芯片构架厂商均有各自成熟的隔离技术，在物联网领域，业界最常见的是基于 Arm TrustZone 的隔离技术和基于 RISC-V 的物理内存保护机制(Physical Memory Protection，PMP)，这两种设计方案均能实现对关键资源的隔离与保护。

TEE 为开放系统提供一系列安全功能，通常包括：

基础服务	功能说明
安全存储	为数据提供加密存储或可信存储区域，保证数据的机密性和完

	整性，防止数据被非法篡改或非法获取。加密密钥动态生成且不出可信安全域。
安全加解密	可支持国际或国密算法，包括对称密码算法、公钥密码算法、散列密码算法等。可支持芯片内置加解密引擎或外接加解密硬件模块。
安全时间	可提供不能被非法篡改的时间接口，仅存于安全可信环境下，主要用于安全日志、视频数据等可审计的场景。
真随机数	可支持在可信环境下获取硬件产生的随机源，且在开放系统无法获取此随机源。主要用于密码算法的随机因子。
安全校验	检查可信应用启动时的完整性和真实性，确保未被非法篡改。
密钥管理	为每个可信应用随机产生共享密钥，保护各可信应用的数据，且可信应用间互不可见密钥。
可信根	绑定硬件唯一标识信息，运行阶段生成。在可信环境下可以获取此信息，且各设备都不同且唯一。
安全驱动	外部设备仅允许 TEE 侧访问（例 SPI、UART、I2C 等），而开放系统无法访问。

基于上述安全功能，可开发各种安全业务（例：安全通信、设备认证、安全升级等）。开发者可根据实际业务的安全需求，可定制开发可信应用来保护设备的安全资产。

3.6. 日志审计及安全态势感知技术

在物联网操作系统中，系统日志是一个重要组成部分，用户可以通过日志来检查错误发生的原因，或者受到攻击时攻击者留下的痕迹。日志主要的功能包括审计和监测，可以实时地监测系统状态、监测和追踪侵入者等。《网络安全等级保护安全设计》中要求：应提供安全审计机制，记录系统的相关安全事件。审计记录包括安全事件的主体、客体、时间、类型和结果等内容。该机制应提供审计记录查询、分类和存储保护，并可由安全管理中心管理。

物联网操作系统的日志审计相对于其他系统场景，有如下特征:

1、审计日志覆盖范围更广。需要覆盖从设备到应用全系统领域，大致可分为：

(1) 设备日志。如感知设备身份鉴别、访问控制和设备状态、设备数据信息安全管理等；
(2) 系统日志。如系统运行、告警、错误信息、关键状态监控信息等；
(3) 运维日志。如用户登录、用户操作、系统更改、异常事件等。

日志存储位置和上报策略需要满足硬件资源限制，日志存储时限需要满足

《网络安全等级保护基本要求》中的约束。

2、日志审计策略需要有很强的及时性。需要及时将安全事件上报，并有一定智能化自处理能力，对重大安全威胁进行实时的防范。

3、日志审计需要具备多层次性。根据物联网设备的资源限制和所处位置，合理存储和逐级上报相应日志信息，形成一套从设备侧到网关侧再到云平台的体系化日志审计系统。

由于物联网场景的如上特征，需要建立统一高效的安全风险识别和通报机制、威胁情报共享机制、安全事件研判处置机制，准确把握网络安全风险发生的规律、动向、趋势，这就需要物联网安全态势感知技术。

物联网安全态势感知技术通过收集物联网环境中综合、全面的安全要素并进行数据融合后，对物联网的安全态势有宏观、全面的认知，并且能对系统的安全趋势进行预测，是保障系统安全的有效手段。态势感知模型是组建一个良好的态势感知系统的核心，为后续顺利地进行态势的提取、评估、预测等工作提供基础。经典的态势感知模型包括 Endsley 的三层模型、数据融合模型和 OODA 控制循环模型。近几年对传统的态势感知模型的融合和发展，形成了多种不同场景下的网络安全态势感知模型，在动态循环、可视化、自动化等角度进行了补充，并根据不同应用场景的需求，实现对模型的丰富和细化。物联网安全态势感知的研究现在仍然处于发展阶段，还有很多问题需要完善和解决，但随着相关技术和研究的不断完善，安全态势感知技术的应用领域将得到不断扩展，与大数据、云计算、物联网和人工智能等新技术的结合将更加紧密。

目前安全态势感知技术在物联网操作系统领域有一些参考实现案例，如图 6 所示。通过集成在物联网操作系统中的安全态势感知模块（安全 SDK）和安全态势感知平台协同，提供一套解决方案。

图 6 物联网安全态势感知技术参考实现方案

安全态势感知平台的实现有如下关键技术：

1、可扩展且开放的大数据平台架构。采用灵活先进的柔性平设计，可以满足各种不同规模的部署场景，支持快速扩展。

2、资产生命周期管理技术。平台可结合漏扫、威胁与情报系统、全流量检测系统、终端安全管理系统和其它外部数据，实现内部网络、外部网络的各类型

资产的安全管理。

3、基于智能决策推理引擎的风险评估和预测技术。基于资产、威胁、漏洞和情报综合进行威胁判定和深度关联分析，综合考虑资产重要性、开放端口/服务/安装中间件、资产漏洞信息、资产上发生的威胁等级、结果、影响，进行威胁与攻击事件的判定。

4、威胁全过程管控的威胁研判体系框架。综合探针、平台全部威胁检测规则，体系化梳理规则可信度、规则热度、风险等级、ATT&CK 战术标记能力，并结合日常运营、攻防演练进行持续化运营和优化，提供高可信+高频+风险等级较高的内置运营场景能力。

5、基于线索式的威胁狩猎与溯源取证技术。平台提供针对可疑线索（IP、域名、MD5 等）关联其上下文，过滤误报，补齐遗漏信息，展示攻击/时间的维度下的攻击过程。

6、安全编排与自动化响应处置 SOAR。平台通过可视化编排将人、安全技术、流程进行深度融合；通过 Playbook 剧本串并联构建安全事件处置的工作流，自动化触发不同安全设备执行响应动作。

7、用户行为分析技术（UEBA）。平台提供用户异常行为分析能力，基于海量的数据，使用高级分析方法和机器学习的模型，对用户和实体(例如 IP 地址、应用、设备和网络等)的行为进行评估、关联并建立基线，以发现内部威胁以及外部入侵行为。

安全态势感知模块的实现有如下关键技术：

1、网络数据采集技术：采集网络通信的五元组信息及 DNS 信息，并上传到云端态势平台处理。

2、本地进程信息采集技术：本地数据以脚本形式采集并保存文件，分别对 CPU 信息、内存信息、进程信息、文件信息、设备信息的进行采集。

3、基线检测技术：订阅基线，获取平台下发的基线（CPU/内存/进程/文件/访问/DNS 策略），对终端行为进行实时监控。

4、入侵检测技术：内置 IDS 检测引擎，针对物联网入侵行为进行检测，覆盖病毒、蠕虫、间谍软件、木马后门、扫描探测、暴力破解等行为。

5、网络黑白名单检测技术：获取平台下发的网络访问控制策略，结合异常行为监控引擎对异常访问事件进行实时监控。

6、进程黑白名单检测技术：获取平台下发的进程控制策略，结合进程监控引擎对异常进程事件进行实时监控。

7、文件黑白名单检测技术：获取平台下发的文件控制策略，结合文件监控引擎对异常文件事件进行实时监控。

8、DNS 黑白名单检测技术：获取平台下发的 DNS 控制策略，结合 DNS 监

控引擎对异常 DNS 事件进行实时监控。

9、响应处置技术：结合基线检测、入侵检测、黑白名单等检测技术，利用处置引擎可进行以下动作：异常进程阻断、恶意文件查杀、非法网络行为阻断、异常 DNS 行为阻断、入侵行为阻断等。

3.7. 系统升级安全技术

物联网操作系统所部署的物联网设备会散布在不同的地方，通常无法近距离地进行直接维护和更新，所以物联网操作系统能够持续不断地进行可靠安全的远程升级/更新是让物联网设备及操作系统安全的一个基础功能。

系统远程升级/更新的安全技术主要涉及更新来源可靠、完整性、机密性、更新失败的恢复、防止回滚等。

1、更新来源可靠：更新来源可靠可以通过公钥签名技术实现。典型的实现路径可以是，当物联网操作系统去服务器检查到可用更新并获取到下载链接后，同时获取更新内容的签名值。利用预置在物联网操作系统内的公钥去验证下载内容的签名值是否正确，保证恶意攻击者无法对物联网操作系统的更新进行破坏。为了进一步保证更新来源的可靠，可以在检查更新服务器和更新下载服务器上均部署 HTTPS 服务。

2、完整性：在保证更新来源可靠的措施公钥签名技术中，通常已经包含了完整性检查，比如签名值通常是对下载内容摘要值或哈希值的签名。

3、机密性：为了减少物联网操作系统被攻击的风险，对于闭源物联网操作系统来说，可以在下载更新时增加对称加密技术的保护措施。

4、更新失败可恢复：即使更新内容被可靠、完整地被下载到物联网设备中，在切换到高版本操作系统时，也可能发生各种未知的错误导致切换失败。当切换到高版本失败时，应该能够回退到升级前的旧版本并且可以正常运行。这可以采用 AB 区乒乓升级的方式来保证更新失败时可以恢复。即有两个分区可以存储完整的物联网操作系统文件，而物联网操作系统正常工作时只使用其中一个分区。假定当前物联网操作系统是存储在 A 区，当有可用更新时，将更新下载到 B 区。如果切换到 B 区的高版本失败，A 区的操作系统仍然是完整可用的，这样使用 A 区的操作系统文件即可恢复物联网操作系统。如果切换成功，则以后使用的就是 B 区的操作系统文件。后续如果再有新的更新，则使用 A 区来下载更新保证更新失败时可以恢复。因此，A 区和 B 区之间反复切换的乒乓升级过程是可以保证更新失败时可恢复的。

5、防止回滚：防止回滚的功能是防止物联网操作系统回退到有安全漏洞的低版本。这个属于高级的物联网操作系统更新安全技术。这个功能如果想安全实现，需要硬件上例如 TEE 等物理可信根的支持。即在物理可信根中保存当前版

本的版本信息，如果启动时发现将要运行的版本低于物理可信根中保存的版本，则操作系统启动失败。正常更新时，如果高版本操作系统可用，则需要及时更新物理可信根中保存的版本信息进行同步。

3.8. 资源竞争安全技术

在资源丰富型物联网操作系统中，当多个应用程序同时对某项内核资源（例如内存、存储空间、信号量、消息队列等）进行调用时，会造成资源不断被修改、资源冲突、资源不可用问题，即资源竞争。资源竞争会对物联网操作系统造成严重的安全问题，例如异常应用可能会占用大量系统资源，影响正常应用的运行，甚至造成物联网设备的宕机。

资源竞争安全技术可有效解决上述问题，此技术可实现对应用的内存大小、磁盘存储空间、信号量、消息队列等资源的配额和优先级上限的配置，对系统可用资源进行配额和操作进行权限检查，避免异常应用程序耗尽系统资源引起其他任务出错。此技术应用在工业安全容器中，可以有效提升设备系统的安全性。

资源竞争安全技术在系统调用层进行资源配额控制，当容器创建时，通过配置文件对其申请的资源上限进行配置，并将配置内容存储在容器控制块中，当系统访问产生资源需求时，产生的需求会通过系统调用层进行比较，当发现超越资源配额上限的时候，系统访问返回失败。

1、内存配额：内存用于暂时存放 CPU 中的运算数据，与外部存储器交换的数据，通常物联网操作系统内存大小主要依据硬件的规格型号，多个程序共享同一个内存，存在资源浪费和互相影响的问题。针对这一普遍问题，资源竞争安全技术对内存进行配额设计，可通过配置文件，对各个应用的内存资源进行分配，设置内存大小，保证当前系统中的程序占用内存不超过配额，避免单个应用程序占用大量资源。

2、文件系统容量配额：文件系统容量也称作磁盘容量，资源竞争安全技术可以通过配置文件，实现对磁盘容量的配额控制，设置磁盘容量大小，分配指定大小的磁盘容量给线程，进行数据的存储和读取，保证当前系统中的程序占用磁盘容量不超过配额。此外，系统会在每次分配和释放空间的时候进行记录磁盘空间使用的详细情况，通过相应的命令可以查看到应用被分配的文件系统磁盘空间配额、文件系统磁盘空间被占用的峰值、当前时刻文件系统磁盘空间使用的情况。

3、信号量配额：信号量是在多线程环境下使用的一种资源，用来保证两个或多个关键代码段不被并发调用。资源竞争安全技术支持丰富的线程间通信信号量，包括二进制信号量，计数信号量，互斥信号量，POSIX 信号量。可通过配置文件，实现对系统的信号量数量进行配置，设置信号量数量，保证程序使用的信号量不超过配额。

4、消息队列配额：消息队列是在消息的传输过程中保存消息的容器，消息队列管理器在将消息从源中继到目标时充当中间人。资源竞争安全技术对消息队列数量进行配置，使当前系统中的各个程序使用的消息队列不超过设置的数量。

5、任务优先级上限配额：资源丰富型物联网操作系统支持任务优先级，优先级是操作系统在处理多个作业任务时，决定各个作业任务接受系统资源的优先等级的参数。资源竞争安全技术可以通过配置文件，实现对应用的任务优先级进行配置，设置最高优先级和最低优先级，保证当前系统中各个线程的优先级在其允许的范围内。

4. 物联网操作系统全生命周期中的安全指导

4.1. 安全设计

在物联网产品设计之初，需要根据物联网产品对应的业务场景和功能引入物联网设备的安全需求，并在物联网操作系统设计过程中通盘考虑以下安全机制以满足安全需求。

1、信任控制：物联网操作系统运行中的相关模块和程序需要经过原厂签名，保证运行的程序可信。针对代码的存储、托管、编译过程以及相关的硬件模块也需要进行周期性的检测和评估，实现信任控制，保证物联网设备没有恶意代码/模块的注入。

2、故障容错：为了避免物联网操作系统在一些特性条件下出现系统故障以及系统中的服务进程僵死或异常退出等情况，除了在进行代码开发过程中多考虑异常情况外，可以考虑利用后台守护进程监控核心服务出现异常时进行核心服务功能恢复，以及引入看门狗硬件电路检测操作系统的异常情况并进行系统功能恢复。

3、最小授权：对物联网设备及操作系统的操作人员进行授权时，应采取最小授权原则。只赋予操作人员执行业务操作所必须的最少权限，并且在限定的时间内完成操作，同时严格限制非必要服务端口的对外暴露。

4、权限分离：应使用多个特权条件来限制用户访问某个对象。

5、纵深防御：建议使用多重防御策略来提高系统安全水平，分别在内核层、服务层、应用层以及数据层等实施安全控制措施，建立纵深防御体系。

6、简洁性：尽量避免把目前并不需要的功能加到信息系统中来，减少错误

发生的机率。开发过程中，代码越简洁，漏洞出现的可能性就会越小。

7、最少共用：避免多个主体通过共享机制使用同一资源。

8、单元分割：单元模块设计应该松散耦合，可独立部署，增加重用性。

9、完全检测：对用户输入系统的任何数据都应当进行合法性校验，必要时还应在系统处理过程中和输出时也进行数据合法性检查。

10、身份鉴别：可以参考物联网操作系统关键安全技术章节中的身份鉴别技术。

4.2. 安全实现

“安全左移”是指将安全保证措施从交付流程的右侧（结束阶段）移至左侧（开始阶段）。详细来讲，针对安全工作的阶段左移，需要在物联网操作系统开发的初期就介入进来：从安全编码原则、软件安全编译、代码安全审计等多个方面进行安全能力内建，例如将安全编码原则形成规范、将安全编译工具集成到持续集成和持续部署流程中、将代码审计和安全漏洞结果导入到缺陷管理工具中等等，由此顺利衔接安全与研发相关工具及流程。

1、安全编码原则

经过安全设计后，通常在开发阶段，会选择多种编程语言对物联网操作系统进行开发。而在程序编码中必须要制定统一、符合标准的编写规范，以保证程序的可读性、易维护性，提高程序运行效率和稳定性。

一些安全组织和企业已经公开了一些编码标准和规范，可以提供参考。如CERT 发布了有关 C、C++、Java 等语言的著名安全编码标准，OWASP 发布了《OWASP 安全编码规范快速参考指南》，MISRA 发布了 MISRA C 编码规范等。根据物联网操作系统的特定需求，结合标准的安全编码规范来制定适合自己的安全编码规范。

2、软件安全编译

在代码编译阶段，应使用安全编译技术来提高编码的安全水平，即采用最新的集成编译环境，通过这些编译环境提供的安全编译选项和安全编译机制来保护物联网操作系统代码的安全性。同时需要考察软件编译的目标环境，尽量使用有最新版本补丁的目标环境进行编译。

3、代码安全审核

为保障安全编程规范的执行，以及在开发阶段由开发小组发现安全问题，需要采用源代码审核的技术手段予以保障。通过分析或检查源代码中的语法、结构、过程、接口等来检查程序的正确性，发现源代码中可能导致安全的薄弱之处，找出代码隐藏的错误和缺陷，如高风险的资源管理、组件间的不安全交互、防御渗

透等。

源代码审核工具包括静态代码审计 SAST 和软件成分分析 SCA。静态代码审计可以在创建阶段及验证阶段部署使用，该方法能够通过分析应用程序的源代码、中间表示文件或二进制文件等来检测潜在的安全漏洞。软件组成分析针对第三方开源软件以及商业软件涉及的各种源码、模块、框架和库进行分析、清点和识别，得到开源软件的组件及其构成和依赖关系，能够识别出已知的安全漏洞或者潜在的许可证授权问题。

目前已经有很多源代码审核工具可供使用，可以分为开源工具和商业工具两类。开源工具中比较有名的包括 BOON、Cqual、Xg++和 FindBugs 等，商业工具中国外厂商有 Fortify、Coverity、CheckMax，国内厂商包括奇安信、悬镜安全、绿盟科技等。

4.3. 安全测试

在物联网操作系统安全开发流程全生命周期中，对物联网操作系统进行安全测试并找到漏洞，然后进行修补优化，可有效降低后期的维护成本。良好的产品防御能力使设备在现网应用不被非法入侵，保证产品的稳定运行。

1、静态代码分析

在开发阶段，需要对源代码进行审计，比如：编码规则、语义缺陷等，找出并修复代码中的各种可能影响系统安全的潜在风险，通过提高代码的自身质量，达到降低物联网操作系统风险的目的。但是代码量比较大，一般会借助静态代码分析类工具，对代码进行自动检测，生成代码安全审计报告，再由人工进行复审。

2、固件包分析

在持续集成阶段，对生成的固件镜像包逆向分析，提取文件系统。对文件系统的每一个文件进行遍历扫描，识别危险的可执行文件、动态链接库、敏感信息文件、关键配置文件、组件等文件类型，判断是否有网络操作等高风险行为，以及是否有缓存溢出、命令注入等高风险。通过与 CVE 漏洞库匹配，分析出有漏洞的组件，发现潜在漏洞。

3、安全动态分析

固件镜像包导入到产品后，对产品整机进行安全需求和功能的验证，通过系统漏洞扫描工具或手工测试的方法监控产品的运行状态或异常行为（如多余的开放端口、系统暴露的已知漏洞、运行的服务安全、敏感数据安全等），发现物联网操作系统运行中的潜在安全问题。

4、模糊测试

模糊测试主要通过向物联网操作系统输入无效数据，以期望触发错误条件或

引起产品的故障，这些错误条件可以指导我们找出那些可挖掘的安全漏洞。模糊测试主要通过以下三种方式：

(1) 借助模糊测试工具触发
(2) 手工编写脚本
(3) 使用发包工具进行组包攻击

5、渗透测试

渗透测试是使用非常规的方法和发散性思维对物联网操作系统进行深入探索和测试，以发现常规测试无法发现的漏洞。常见的渗透测试过程从信息收集开始，收集端口信息、系统信息、组件信息等，查看是否有暴露的漏洞；然后就是漏洞尝试利用，进而去提取系统 shell 权限等。

表 5 是根据《YD/B 173-2017 物联网终端嵌入式操作系统安全技术要求》标准整理的安全测试要点，供参考。

表 3 物联网操作系统安全测试点

测试项	测试子项	测试点
接入安全	网络接入认证	验证系统接入网络时采用的身份认证及鉴别机制
	网络访问控制	验证系统访问网络时采用的双向身份认证机制
通信安全	传输完整性	通过校验工具查看传输前后的数据保持一致
	传输保密性	通过流量分析数据在传输过程中未采用明文传输
	抗重放攻击	测试系统在数据通信时，抓取报文后重放报文，系统不响应或者提示无效报文
系统安全	用户身份鉴别	用户标识的唯一性
		用户登录的身份鉴别
	访问控制	操作系统具备访问控制机制，根据系统管理员设置的访问控制策略，系统中所有主体、客体都应遵循访问控制机制
	系统安全审计	审计范围应覆盖到每个系统用户
		审计内容应包括重要用户行为、系统资源的异常使用等重要的安全相关事件
		审计记录应包括事件的日期、时间、类型、主体标识、客体标识和结果等
		保护操作系统审计记录，避免被删除、修改或覆盖
	系统安全隔离	通过命令验证终端对操作系统资源和各类数据进行安全隔离，存储空间进行划分
	资源访问控制	验证系统中客体的访问属性读、写、执行等访问权限设置
		客体仅允许客体拥有者对其访问权限进行设置，且客体拥有者无法把客体的控制权分配给其他主体
	升级更新机制	升级通道加密
		升级镜像包加密

		升级失败时系统可回退可用
数据安全	数据完整性	存储数据内容进行完整性保护和验证，检测到完整性错误时采用必要的恢复措施
	数据可用性	验证系统重要数据有备份机制，删除之后仍然可以正常恢复
	数据机密性	对鉴别信息、重要业务数据等敏感内容进行加密存储验证
个人信息安全	个人信息检测	操作系统中不应该存储个人相关信息数据，除非有客户授权
	个人信息删除	检查个人信息彻底删除，且不可恢复

4.4. 安全运维

物联网操作系统的安全运维要从身份标识、访问控制、系统检测处理机制、漏洞管理、网络隔离、数据安全、日志审计等方面进行要求。

1、身份标识要求：采用各类工具和技术，对物联网操作系统的身份标识进行记录、分析和告警；通过制定流程和制度，建立系统身份标识管理体系，保障物联网操作系统的身份标识的唯一性，减少故障或风险的产生。

2、访问控制要求：基于访问的安全要求，应建立物联网操作系统的访问控制策略，形成文件并进行评审。每季度审核所有系统账号，建议定期更新，口令符合安全复杂度的要求。建议不要采用互联网远程维护访问方式。

3、系统检测处理机制要求：应定期对物联网操作系统指标进行检测，检测内容包括但不限于系统版本、配置参数、资源占用情况、系统备份机制、数据备份容灾功能、补丁更新、安全防护、漏洞扫描、病毒库更新等，当发现设备异常，启动系统异常处理机制。当检测到系统异常，实时通知管理员安全风险的细节并提供处理建议，阻止可能存在的安全威胁。

4、漏洞管理要求：全面了解物联网操作系统存在的漏洞，组织评价这些漏洞的可利用性，尽快对高危漏洞进行补丁加固。并留有补丁升级记录，便于查阅追溯。可通过下面两种方式获取漏洞。

(1) 定期对物联网操作系统进行漏洞扫描（扫描前要更新扫描器的漏洞插件库），以发现最新的漏洞；

(2) 通过官方渠道及时了解使用的物联网操作系统存在的漏洞。

5、网络隔离要求：物联网设备及操作系统应采取网络隔离技术，部署一定的安全防护系统如：应用防火墙、入侵防御系统等，实现终端和网络层的安全隔离，降低公网入侵风险。建立端口管理体系，严格控制端口的打开与关闭，并建立 ACL 列表，只允许经过授权的服务请求。

6、运维数据安全要求：主要包括数据存储安全和数据备份安全要求。

(1) 物联网操作系统产生的重要数据应加密存储，应生成数据存储日志以便运维管理，包括但不限于数据存储日期，数据加密方式，数据敏感程度，存储位置，角色访问权限等。

(2) 物联网操作系统产生的关键数据要有备份机制。在数据丢失、损坏、删除后供恢复使用。

7、日志审计：在物联网操作系统层、数据库层、应用层建立日志记录功能，日志能够记录关联操作用户的身份、操作行为等关键要素，且日志可以保存一定的时长并不可被破坏。

5. 物联网操作系统安全技术应用实例

5.1. 工业安全容器

翼辉信息于 2019 年推出了基于其自主操作系统的工业安全容器（ECS），创新性地将容器技术引入工业领域，采用资源竞争安全技术，用以解决物联网嵌入式设备容易受到来自不可信赖或恶意程序的攻击以及应用资源竞争等问题。

工业安全容器技术支持将应用环境整体打包成为一个标准化单元，实现开发、交付、部署环境的一致性，使其免受外在环境差异的影响，有助于减少相同硬件设施上运行不同软件时的冲突。

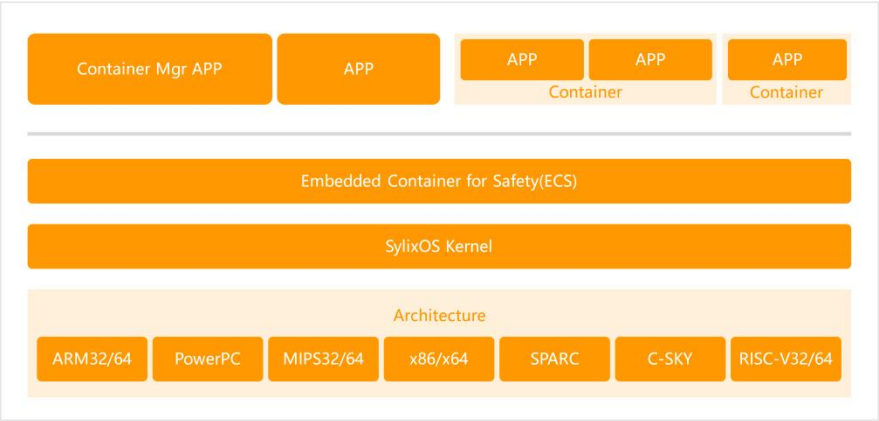

图 7 工业安全容器架构

工业安全容器技术可在物联网操作系统和容器内的应用程序之间提供一道安全的隔离边界，可实现系统资源隔离和资源配额管理，这种新型的沙箱技术能

为容器提供安全隔离机制，同时比虚拟机更轻量级、更高效。

容器支持全方位的资源隔离技术，可对内存地址空间、文件系统、设备访问、环境变量、POSIX 命名空间、AF_UNIX 通信域、进程间通信 IPC 等资源进行隔离，确保容器具有完整独立的运行环境，免受外在环境差异的影响。

容器支持资源竞争安全技术，可对容器的 CPU 运行时间、内存大小、内核对象、磁盘存储空间、信号量、消息队列、任务优先级等资源的配额和 shell 命令权限、设备访问权限进行配置，对容器可用资源进行配额和操作进行权限检查，提升系统整体的安全性。

容器支持轻量级实现，对 CPU 和内存消耗极少，系统镜像可控制在 6MB 以内，为工业场景的应用提供了更加安全可靠的系统解决方案，满足其对性能、实时性、安全性、可靠性的高标准要求。

5.2. 平台安全架构

5.2.1. PSA简介

在针对万亿级别物联网设备及其产生的海量数据的安全需求的愿景下，Arm 公司在 2017 年 10 月对外发布了 IoT 领域的安全框架 - Platform Security Architecture（平台安全架构，以下简称 PSA）。其目标是为 IoT 产业链，从芯片制造商到设备应用开发者的各个环节，提供一套完整的安全指导。确保接入物联网的设备及其输出的数据都是可信的。

PSA 提供了从安全分析、软硬件架构规范、参考实现、再到认证的完整流程：

1、安全分析：通过对目标系统进行威胁建模，明确在设计时所必须的安全需求。Arm 提供了三个常见 IoT 用例的参考威胁模型和安全分析（Threat Models and Security Analyses，TMSA），分别对应物品追踪器、智能水表以及网络摄像头。开发者可以基于这些文档开展针对目标系统的具体安全分析，明确自己的安全需求。

2、软硬件架构规范：Arm 提供了一系列软硬件架构规范文档，用来帮助系统设计者实现上述安全需求。针对 IoT 领域主要的文档包括：

(1) 固件框架（Firmware Framework for M）：定义了运行在 Arm® Cortex®-M 系列处理器上的安全应用程序的标准编程环境及基础的根信任。

(2) 可信基础系统架构-TBSA（Trusted Base System Architecture for M）：提供了针对 Armv8-M 架构处理器的硬件要求规范，以及针对不支持 TrustZone 的 Armv6-M 和 Armv7-M 架构处理器的最佳实践建议。

(3) 安全模型（Security Model）：提供安全设计的顶层要求。概括了设计

具有已知安全属性的产品的关键目标。

(4) 平台安全启动指南（Platform Security Boot Guide）：提供了针对固件启动及更新的技术要求。

(5) 平台安全要求（Platform Security Requirements）：规定了对 SoC（System-on-Chips）预期的最低安全要求。

(6) 基于认证的调试访问控制规范（Authenticated Debug Access Control Specification，ADAC）：定义了一种可扩展的方法，用来在调试过程中引入强身份验证。

3、参考实现：Arm 提供了一套开源的 PSA 固件参考实现。给开发者提供了一个符合 PSA 规范的可信代码参考，以及对接底层可信根硬件的安全 API 接口。

(1) 参考实现：针对 IoT 领域 Arm 提供 TF-M 作为 PSA 固件参考实现，参考下一小节对 TF-M 的介绍。

(2) PSA 功能 API（PSA Functional APIs）：目前定义了针对密码学（Cryptography），安全存储（Secure Storage），设备证明（Attestation）及固件升级（Firmware Update）相关的 API。

(3) API 测试程序（API test suite）：用来测试各 API 接口有没有被正确实现。

4、认证：提供基于安全实验室评估的安全认证。PSA 认证目前包含两部分内容[7]：

(1) 功能 API 认证：该认证的目的是确保 PSA 定义的 API 被正确的实现并检查其一致性。

(2) 安全认证：提供了三个级别的安全认证对应到不同场景的安全需求。

表 4 PSA 安全认证等级概述

认证级别	第 1 级	第 2 级	第 3 级
稳健性级别	确保 RoT、OS 和设备的安全模型目标	基于实验室的 PSA-RoT 防软件攻击和“轻量级”硬件攻击的评估	可以防御额外的大量软件和硬件攻击
认证过程	实验室检查问卷	实验室评估、白盒测试	实验室根据更高级别的保护轮廓（PP）要求，进行测试
认证结果	PSACertified.org 上的证书	PSACertified.org 上的证书	PSACertified.org 上的证书

上述规范文档资料可以从 Arm 官网免费下载：

https://developer.arm.com/architectures/architecture-security-features/platform-security。

PSA 认证相关内容可参阅：https://www.psacertified.org/。

5.2.2. TF-M简介

TF-M 全称为 Trusted Firmware M，是 Arm 创建的符合 PSA 规范的开源固件

实现。目前由 Trusted Firmware 社区管理。其使用 BSD-3 开源许可协议。支持 PSA 定义的 Firmware Framework API 及 Functional API，并且随着 PSA 规范一起演进。目前通过 PSA 认证的系统普遍使用了 TF-M 作为安全侧固件。

TF-M 不仅支持拥有 TrustZone 技术的 Armv8-M 架构处理器，而且也支持使用双 Arm® Cortex®-M 核处理器或者 Arm® Cortex®-A 核+Cortex®-M 核的多处理器系统。

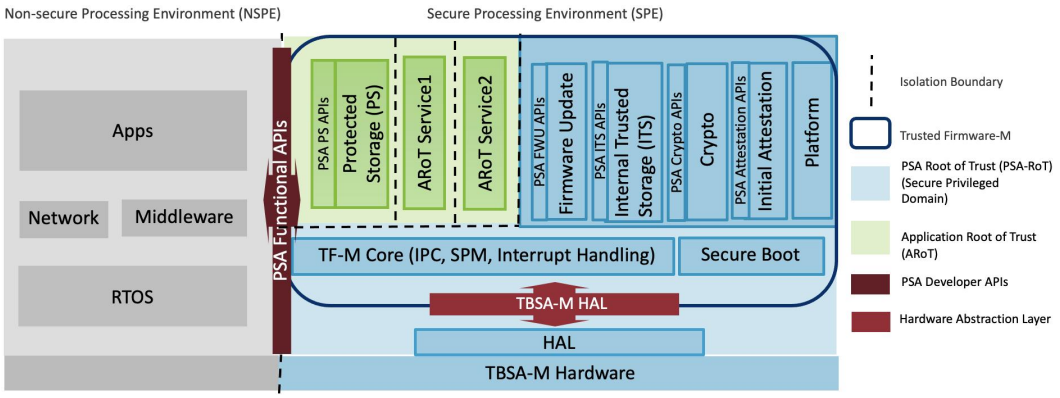

图 8 运行 TF-M 的典型系统模块框图

如图 8 所示，TF-M 主要提供了这些功能特性：

1、安全启动：TF-M 目前默认适配了 MCUBoot 作为 Bootloader 来进行安全启动引导。在 TF-M 及非安全侧的用户镜像文件加载和执行前，会验证签名，确保镜像文件的合法性及完整性。同时支持镜像文件加密和防回滚。通过简单的适配用户即可支持自己的 Bootloader 或者 RomCode。

2、安全存储：支持使用芯片内部安全 Flash 或者外部 Flash 进行安全存储。可以对存储内容进行加密、完整性验证及防回滚保护。

3、加解密服务：默认使用 mbed TLS 作为加解密引擎，提供各种基于密码学的服务，比如哈希、对称加密、非对称加密、签名及验证等。用户可适配自己的底层加解密库。

4、OTA 镜像升级服务：与 Bootloader 配合提供验证及更新系统镜像的服务。

5、设备证明服务：验证方通过挑战设备，获得设备的相关信息。该信息及挑战的随机数通过设备私钥签名。典型的设备信息包括：支持的安全功能，组件版本、安全生命周期等。

更多关于 TF-M 的细节可以在 TF-M 帮助文档[8]中找到。

5.3. 嵌入式防火墙

基于网络通信安全技术，翼辉信息研发了一款适用于物联网设备及操作系统的嵌入式防火墙，能有效防御常见的网络攻击，包括网络风暴攻击，重放攻击，

ARP 欺骗攻击，SYN 泛洪攻击等，以保护物联网设备及系统安全。

此嵌入式防火墙采用双层设计，上层实现检测管理，下层实现报文过滤，如图 9 所示。基于翼辉原创的 SylixOS 内核，这两个部分可配置各自的内存管理单元减少对系统资源的占用，整体采用支持轻量化设计，防火墙镜像不大于 220K。

防火墙内部包括四个防御模块，网络风暴防御模块，重放攻击防御模块，ARP 欺骗防御模块，及 SYN 泛洪防御模块。四种防御模块具体功能如下：

1、网络风暴防御模块：自动识别产生风暴的设备和风暴报文类型；实时监测每一种类型报文流量，支持动态配置监测参数；对问题设备采取拉黑操作，支持动态配置拉黑时间。

2、重放攻击防御模块：通过随机验证码进行报文唯一性验证，支持验证码产生时间隔间修改；支持局域网和非局域网环境下的重放攻击防御。

3、ARP 欺骗防御模块：自动识别新加入网络的设备信息；自动进行 MAC（Media Access Control Address）与 IP 的绑定；智能识别当前网络设备的 MAC 变化情况。对 MAC、IP 地址发生变化和 ARP 欺骗这两种网络状况，支持快速识别与处理。

4、SYN 泛洪防御模块：自动识别产生 SYN 泛洪攻击的设备；自动识别 SYN 泛洪的起始与结束；实时监测 SYN 报文流量，支持动态配置监测参数；产生 SYN 泛洪时，采取白名单通信机制。

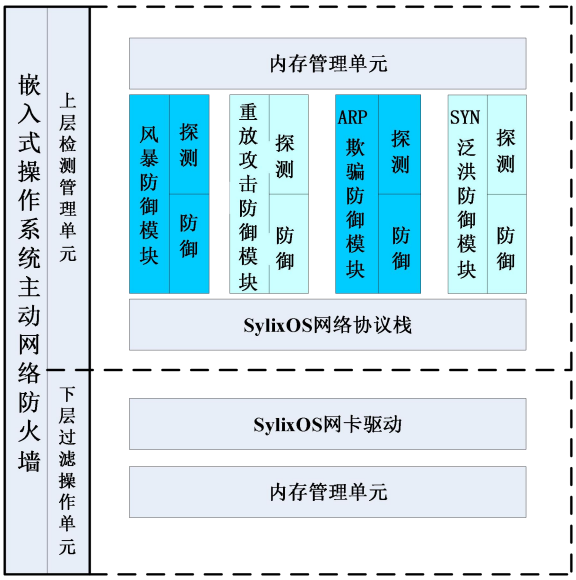

图 9 嵌入式防火墙结构图

5.4. 轻量级传输层安全协议

OneTLS 是中移物联网研发的一款轻量级 TLS 安全通信协议栈，实现了最新的(D)TLS1.3 协议，小巧灵活且易于使用，方便开发人员快速地集成进物联网产

品中。OneTLS 可以根据实际情况灵活地裁剪，以降低对硬件资源的消耗。

OneTLS 具有以下技术特点：

1、支持最新的 TLS1.3（RFC 8446）和 DTLS1.3（RFC 9147）标准，取消了 MD5、3DES、RC4 等安全性较低的密码算法，密钥交换支持前向安全，握手协议和记录协议的密钥实现分离，握手消息进行了加密操作，安全性大大提高；

2、支持 PSK 密钥协商及身份认证机制，可实现 0-RTT 数据传输，节省了连接建立时的交互次数，实现更快的访问速度；

3、支持商密算法，实现了 IETF 标准商密套件（RFC 8998 ShangMi (SM) Cipher Suites for TLS 1.3），同时支持 PSA Cryptography API，可通过统一接口完成软件计算、MCU 内部加密引擎或片外安全芯片的密码计算调用；

4、资源占用低至 15KB，适用于存储资源受限的场景，有效降低物联网设备安全通信的硬件门槛和成本。

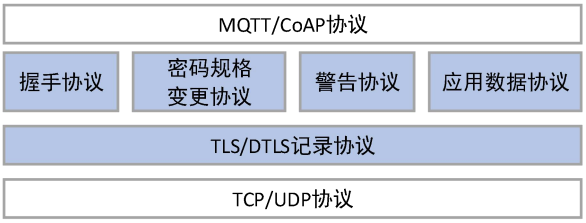

图 10 OneTLS 分层结构图

6. 建议及展望

物联网感知控制域复杂多变的环境使得物联网设备及操作系统面临着严峻的安全挑战，因而有着多方面的安全需求。但另一方面物联网设备异构多样和资源有限的现实状况却使得物联网设备及操作系统难以承载复杂的安全技术和措施。面对这些挑战，我们希望与合作伙伴通力合作，围绕物联网操作系统安全开展多方面的工作，共同推动物联网设备及操作系统安全技术发展和应用：

1、共同完善物联网操作系统安全技术体系

持续完善物联网操作系统安全技术体系建设，定义统一的安全接口和规范，以指导实现物联网操作系统安全子系统的轻量化模块化设计。

2、共同促进物联网软硬安全技术协同发展和应用

在物联网操作系统层面充分利用物联网设备所具有的硬件安全能力，如轻量的物理安全标识、安全存储、安全加密等，以软硬件结合的方式提升物联网设备的安全性和协同性发展。

3、共同加快物联网设备及操作系统内生安全能力的构建

物联网操作系统层面充分支持安全启动、可信执行环境、安全隔离、可信监测等内生安全机制，以构建物联网设备及操作系统内生安全能力的方式来面对复杂多变的物联网安全环境，加快构建物联网设备及操作系统内生安全的能力。

缩略语列表

缩略语	英文全名	中文解释
ADAC	Authenticated Debug Access Control Specification	基于认证的调试访问控制规范
ABAC	Attribute-based Access Control	基于属性的访问控制
AEAD	Authenticated Encryption with Associated Data	用于关联数据的认证加密
AES	Advanced Encryption Standard	高级加密标准
ASLR	Address Space Layout Randomization	地址空间布局随机化
BLE	Bluetooh Low Energy	蓝牙低能耗技术
CBC	Cipher Block Chaining	密码分组链接模式
CCM	Cipher Block Chaining-Message Authentication Code	密码分组链接消息认证码
CFB	Cipher Feedback Mode	加密反馈模式
CNSA	Commercial National Security Algorithm	美国商业国家安全算法
CTR	Counter mode	计数器模式
DAC	Discretionary Access Control	自主访问控制
DDoS	Distributed Denial of Service	分布式拒绝服务
DES	Data Encryption Standard	数据加密标准
DEP/NX	Data Execution Prevention/No-execute	DEP/NX数据执行保护技术
DH	Diffie Hellman Key Exchange Algorithm	迪菲-赫尔曼密钥交换协议/算法
DICE	Device Identifier Composition Engine	设备标识组合引擎
DTLS	Datagram Transport Layer Security	数据包传输层安全协议
ECB	Electronic Code Book	电子密码本模式
ECC	Elliptic Curve Cryptography	椭圆曲线
ECDH	Elliptic Curve Diffie-Hellman	椭圆曲线迪菲-赫尔曼算法
ECDSA	Elliptic Curve Digital Signature Algorithm	椭圆曲线数字签名算法
GCM	Galois Counter Mode	伽罗华计数器模式
GPIO	General-Purpose Input/Output	通用型输入输出
GPRS	General Packet Radio service	通用分组无线业务
HKDF	HMAC-based Extract-and-Expand Key Derivation Function	基于哈希的提取和扩展密钥派生函数

HMAC	Hash-based Message Authentication Code	基于哈希的消息认证码
HSM	Hardware Security Module	硬件安全模块
HSPA	High-Speed Packet Access	高速下行分组接入
IMEI	International Mobile Equipment Identity	国际移动设备识别码
IIoT	Industry Internet of Things	工业物联网
IoT	Internet of Things	物联网
KDF	Key Derivation Function	密钥派生函数
LSM	Linux Security Modules	Linux安全模块
LwM2M	lightweight Machine to Machine	轻量级M2M协议
MAC	Message Authentication Code	消息认证码
MAC	Mandatory Access Control	强制访问控制
MAC	Media Access Control Address	媒体存取控制位址
NFC	Near Field Communication	近场通信
NFV	Network Function Virtualization	网络功能虚拟化
NSA	National Security Agency	美国国家安全局
NSPE	Non-secure Processing Environment	非安全执行环境
OBAC	Object-based Access Control	基于对象的访问控制
OFB	Output Feedback Mode	输出反馈模式
PIE	Position Independent Executables	位置无关可执行文件
PSA	Platform Security Architecture	平台安全架构
PMP	Physical Memory Protection	物理内存保护机制
RBAC	Role-based Access Control	基于角色的访问控制
RCE	Remote Command/Code Execute	远程命令/代码执行
RoT	Root of Trust	信任根
RSA	Rivest-Shamir-Adleman	RSA算法
RTOS	Real-time operating system	实时操作系统
SMACK	Simplified Mandatory Access Control Kernel	简化的强制访问控制内核
SPE	Secure Processing Environment	安全执行环境
SPI	Serial Peripheral Interface	串行外设接口
TBAC	Task-based Access Control	基于任务的访问控制
TBSA	Trusted Base System Architecture	可信基础系统架构
TCG	Trusted Computing Group	可信计算组织
TCM	Trusted Cryptography Module	可信密码模块
TDES	Triple Data Encryption Standard	三重数据加密标准

TEAC	Type Enforcement Access Control	类型增强访问控制机制
TF-M	Trusted Firmware M	符合PSA规范的Arm M系列处理器安全固件
TLS	Transport Layer Security	传输层安全协议
TMSA	Threat Models and Security Analyses	威胁模型和安全分析
TPCM	Trusted Platform Control Module	可信平台控制模块
TPM	Trusted Platform Module	可信平台模块
TRNG	True Random Generator	真随机数发生器
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器
UDS	Unique Device Secret	唯一设备秘密

参考文献

[1] 全球移动经济报告（2021 年），GSMA
[2] 李运喜. 物联网环境下嵌入式操作系统技术特性研究. 航空计算技术，1671-654X（ 2018） 06-0082-04
[3] 物联网白皮书（2020 年），中国信息通信研究院
[4] 腾讯宙斯盾团队 剑圣. 现网发现新型 DVR UDP 反射攻击手法记实，2020
[5] 物联网安全标准化白皮书（2019 年版），全国信息安全标准化委员会通信安全标准工作组
[6] GlobalPlatform Technology Cryptographic Algorithm Recommendations Version 2.0 June 2021 https://globalplatform.org/specs-library/globalplatform-technology-cryptographic-algorithm-recommendations/
[7] Arm Ltd.. PSA 认证：概述[EB/OL]. https://aijishu.com/a/1060000000003070. 2019 年 7 月
[8] TF-M 开发团队. TF-M 帮助文档[EB/OL]. https://tf-m-user-guide.trustedfirmware.org/. 2021 年 10 月