软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(2):641−663 [doi: 10.13328/j.cnki.jos.006154]

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

SG-Edge: 电力物联网可信边缘计算框架关键技术*

杨维永[1,2], 刘 苇[2], 崔恒志[3], 魏兴慎[2], 黄 皓[1], 廖 鹏[2,4], 钱柱中[1], 王元强[2]

[1](南京大学 计算机科学与技术系, 江苏 南京 210023)

[2](南瑞集团有限公司(国网电力科学研究院有限公司), 江苏 南京 210003)

[3](国网江苏省电力有限公司, 江苏 南京 210003)

[4](北京邮电大学 网络空间安全学院, 北京 100876)

通信作者: 刘苇, E-mail: liuwei5@sgepri.sgcc.com.cn

摘 要: 随着国家电网电力物联网的逐步推进, 作为其核心支撑技术的边缘计算框架逐渐成为研究热点. 首先, 总结了物联网和边缘计算框架方面的已有研究工作; 其次, 通过分析电力物联网在业务场景、边缘计算、信息安全等方面的关键技术难题, 提出了一种适应于电力物联网的可信边缘计算框架 SG-Edge; 随后, 结合边缘框架的可信防护关键难题, 给出了硬件可信引导、软件行为动态度量等关键技术方法; 最后, 从业务适应性、安全性以及性能等方面对 SG-Edge 进行了全面评估, 并对未来研究可能面临的挑战进行了展望.

关键词: 物联网; 电力物联网; 边缘计算; 可信计算; 网络安全

中图法分类号: TP393

中文引用格式: 杨维永, 刘苇, 崔恒志, 魏兴慎, 黄皓, 廖鹏, 钱柱中, 王元强. SG-Edge: 电力物联网可信边缘计算框架关键技术. 软件学报, 2022, 33(2): 641−663. http://www.jos.org.cn/1000-9825/6154.htm

英文引用格式: Yang WY, Liu W, Cui HZ, Wei XS, Huang H, Liao P, Qian ZZ, Wang YQ. SG-Edge: Key Technology of Power Internet of Things Trusted Edge Computing Framework. Ruan Jian Xue Bao/Journal of Software, 2022, 33(2): 641−663 (in Chinese). http://www.jos.org.cn/1000-9825/6154.htm

SG-Edge: Key Technology of Power Internet of Things Trusted Edge Computing Framework

YANG Wei-Yong[1,2], LIU Wei[2], CUI Heng-Zhi[3], WEI Xing-Shen[2], HUANG Hao[1], LIAO Peng[2,4], QIAN Zhu-Zhong[1], WANG Yuan-Qiang[2]

[1](Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China)

[2](NARI Group Corporation (State Grid Electric Power Research Institute), Nanjing 210003, China)

[3](State Grid Jiangsu Electric Power Co., Ltd., Nanjing 210003, China)

[4](School of Cyber Science and Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: With the gradual advancement of the State Grid power Internet of Things (IoT), the edge-computing framework as its core support has gradually become a research hot topic. This article firstly summarizes the existing research work on the IoT and edge-computing framework. Secondly, by analyzing the key requirements of the power IoT in business scenarios, edge computing and information security; a way to adapt to the power SG-Edge is proposed, which is a trusted edge-computing framework for the IoT. Then, key technical methods such as hardware trusted boot and dynamic measurement of software behavior are proposed to meet the key challenges of the trusted protection of the edge framework. Finally, the SG-Edge is fully evaluated from the aspects of business adaptability and security, and the possible future research challenges are prospected.

Key words: Internet of Things (IoT); power IoT; edge computing; trusted computing; network security

2019 年, 国家电网公司“两会”做出全面推进“三型两网”建设, 加快打造具有全球竞争力的世界一流能源

* 基金项目: 国家电网总部科技项目(5210ED209Q3U); 智慧物联体系边缘智能计算框架与关键技术
收稿时间: 2019-06-30; 修改时间: 2020-04-02; 采用时间: 2020-09-12

互联网企业的战略部署, 是网络强国战略在公司的具体实践, 是落实中央部署、发挥央企带头作用的重要举措, 是适应内外部形势和挑战的必然要求. 电力物联网, 就是围绕电力系统各环节, 充分应用移动互联、人工智能等现代信息技术、先进通信技术, 实现电力系统各环节万物互联、人机交互, 具有状态全面感知、信息高效处理、应用便捷灵活特征的智慧服务系统. 建设电力物联网, 为电网运行更安全、管理更精益、投资更精准、服务更优质开辟了一条新路, 同时也可以充分发挥电网独特优势, 开拓数字经济这一巨大蓝海市场. 建设电力物联网, 是落实“三型两网、世界一流”战略目标的核心任务.

随着电力物联网的逐步推进, 边缘计算框架逐渐成为其中的研究热点, 边缘计算框架的设计多种多样, 一般包括如下功能: 基于边缘操作系统的资源管理、子设备接入、数据采集与设备控制、安全管理、应用管理以及物联网平台交互等功能. 而根据设计目标和应用部署场景, 一般可以分为“面向物联网的边缘计算、面向边缘云服务的边缘计算、面向云边融合的边缘计算”这 3 类[1]. 其中,电力物联网既有面向物联网的边缘计算的特性, 又有一定的云边融合的应用场景.

为此, 本文设计实现了一套适应电力物联网的可信边缘计算框架 SG-Edge, 满足电力物联网的边缘计算功能要求, 重点通过安全操作、硬件保障技术保障边缘计算安全可信. 框架技术路线如下.

- 在框架技术路线方面, 充分借鉴成熟框架, 满足不断增长的业务需求; 同时, 结合电力安全进行框架设计;
- 在等级保护合规性方面, 充分考虑利用新型硬件的可信引导机制, 实现边缘网关从 CPU 引导、到 BootLoader、到操作系统内核可信验证, 实现主动免疫, 确保固件安全启动, 安全可信升级;
- 在软件行为安全保证方面, 通过 TrustZone 和可信安全模块等方式物理隔离措施保障可信软件基安全, 并且通过流处理实现软件动态行为的判定;
- 在态势感知与威胁监测方面, 结合 KillChain 模型和 ATT&CK 知识库, 构建基于攻击视角的行为分析, 发现异常攻击, 作为可信保证体系的补充, 进一步提升物联网主动防御能力.

本文第 1 节对边缘计算现有的框架及相关工作进行总结, 并标识出各个框架的优缺点. 第 2 节对 SG-Edge 边缘计算框架设计思路进行总体介绍. 第 3 节对 SG-Edge 的可信保护机制关键技术点进行总结. 第 4 节从可行性、性能以及安全性等方面开展对 SG-Edge 的全面评估. 第 5 节总结全文, 并对未来值得关注的研究方向进行初步探讨.

1 前 言

1.1 电力物联网的建立与进展

- 在学术研究方面.

2016 年 10 月, 由 IEEE 和 ACM 正式成立了 IEEE/ACM Symposium on Edge Computing, 组成了由学术界、产业界、政府共同认可的学术论坛, 对边缘计算的应用价值、研究方向开展了研究和讨论[2−5], 这两年尤其关注物联网场景下的性能[6]、安全性[7]、应用场景[8,9]、云边协同[10]以及与 AI 等技术的融合[11−13]. 2018 年 5 月, 边缘计算技术研讨会(SEC China 2018)在西安召开, 众多高校和科研机构互动研讨边缘计算, 进一步梳理开发者需求. 此外, 国内众多学者针对边缘计算场景下的数据模型[14]、计算模型[15]、工业应用[16−19]、网络安全[20,21]等也开展了广泛研究.

- 在标准化方面.

2017 年, IEC 发布了 VEI (vertical edge intelligence)白皮书[13], 介绍了边缘计算对于制造业等垂直行业的重要价值. ISO/IEC 成立边缘计算研究小组. 在 IEEE P2413 物联网框架(standard for an architectural framework for the IoT)中, 边缘计算成为了该框架的重要内涵. 中国通信标准化协会(CCSA)成立了工业互联网特设组(ST8).

- 在产业联盟方面.

2016 年 11 月, 华为、中国电科院、中国信息通信研究院、英特尔、ARM 和软通动力信息技术有限公司

联合倡议发起边缘计算产业联盟. 2019 年 11 月 28 日, 边缘计算产业联盟发布 3 本边缘计算领域的白皮书, 分别为《边缘计算 IT 基础设施白皮书 1.0(2019)》、《运营商边缘计算网络技术白皮书》、《边缘计算安全白皮书》. 2017 年, 在全球性产业组织工业互联网联盟 IIC 的组织下, Edge Computing TG 成立, 定义了部分边缘计算参考框架.

2019 年, 国家电网公司“两会”做出全面推进“三型两网”建设, 加快打造具有全球竞争力的世界一流能源互联网企业的战略部署, 电力行业众多研究者也开始相关的应用研究和实践[22−25]. 南方电网公司提出“透明电网”概念: 把现代信息技术与电网相结合, 在电网上安装小微智能传感器, 让电力系统的各个环节展示出来, 包括电源信息透明、网络信息透明、市场信息透明、设备状态透明、运行状态透明、交易状态透明等等, 形成“透明电网”. 电力物联网具备工业控制系统和工业物联网双重属性, 在确保高安全性和高可靠性的基础上, 提出具有工业物联网特性的 APP 容器化、可信远程升级等运维功能, 快速响应业务需求的目标.

1.2 电力物联网边缘计算框架

以具体的边缘计算框架而言, 面向物联网的边缘计算、面向边缘云服务的边缘计算、面向云边融合的边缘计算是目前主流的边缘计算框架. 其中,

- 面向物联网的边缘计算致力于解决在开发和部署物联网应用的过程中存在的问题, 如设备接入方式多样. 以面向工业物联网边缘计算开发的标准化互操作性框架 EdgeX Foundry[26]为例, 它围绕互操作性组件的生态系统, 提供了极为简化和标准化的工业物联网边缘计算架构; Apache Edgent[27]是一种编程模型和具有微内核风格的运行时边缘框架, 它主要关注如何对来自边缘的数据进行高效的分析处理, 可以加速边缘计算应用在数据分析上的开发过程, 可部署于运行 Java 虚拟机的边缘计算中, 实时分析来自设备的数据, 具有丰富的 API, 切合物联网的实际加速开发需求; Predix[28]面向制造业, 提供开发框架, 支持开放现场协议的接入, 增强了边缘计算的功能, 由合作伙伴开发相应的设备接入和边缘计算的功能;
- 面向边缘云服务的边缘计算主要着眼于优化或重建网络边缘的基础设施, 以实现在网路边缘构建数据中心, 并提供类似云中心服务, 通常见于网络运营商的网络边缘, 如蜂窝网络基站. 代表性的有开放网络基金会(ONF)的 CORD, 它利用软件定义网络、网络功能虚拟化(NFV)云计算技术重构现有网路边缘, CORD 在运营商边缘提供边缘云服务, 对用户而言无需提供计算资源, 搭建平台降低软硬件成本; 此外, Linux 基金会提供了一套面向高性能边缘云的开源项目 Akraino Edge Stack, 致力于开发一整套开源软件栈, 用于优化边缘基础设施的网络构建和管理;
- 面向云边融合的边缘计算, 云计算服务提供商是边缘计算的重要推动者, 基于“云边融合”理念, 致力于将云服务能力拓展到网络边缘, 典型的包括 AWS 的 GreenGrass、百度的 OpenEdge[29]、阿里的 Link IoT Edge 以及 Azure IoT Edge, 旨在混合云和边的边缘计算框架, 拓展云功能到边缘设备以获得低延时. 边缘框架运行于边缘设备上, 往往使用云上相同的编程模型.

不同框架针对边缘计算的理解、方案设计和实现思路各不相同, 框架之间无法实现兼容. 现有的计算框架方面:

- OpenEdge 功能有限, 并且与百度物联网平台绑定比较紧密, 但可借鉴函数计算思想;
- KubeEdge 基于 Kubernetes 技术适配边缘计算, 对平台技术有限制, 与平台紧耦合;
- EdgeX 模块之间解耦, APP 以微服务形式运行, 对 APP 管理也以 REST API 调用的方式实现, 是比较完善的工业物联网解决方案, 但缺乏云边融合以及安全的考虑. EdgeX 仅提供了数据导出的接口, 无法直接与物联管理平台通信, 需要基于交互规范开发与物联管理平台的交互流程. EdgeX 缺乏应用下发、升级、管理、业务 APP 控制、设备管理控制、监视等功能. 同时, EdgeX 缺少安全加固方案, 在安全接入、访问控制、应用命令验证等方面缺少设计.

1.3 面向边缘的安全可信技术

- 物联网及边缘计算安全风险研究

1) 边缘接入方面, 边缘接入不安全的通信协议, 可能存在恶意的边缘节点; 2) 边缘节点本体安全方面, 边缘节点数据易被损毁, 隐私数据保护不足, 不安全的系统与组件, 易发起分布式拒绝服务, 易蔓延 APT 攻击, 硬件安全支持不足; 3) 在边缘管理方面, 身份、凭证和访问管理不足, 账号信息易被劫持, 不安全的接口和 API, 难监管的恶意管理员. 2019 年末, 边缘计算产业联盟《边缘计算安全白皮书》[29]从识别、解释和定位与边缘安全相关的体系结构、设计和技术、从边缘安全的重要性和价值出发, 分析了典型价值场景下边缘安全面临的挑战和需求特征, 并提出了边缘安全的参考框架和确保处理相应安全问题的方法组合. 等级保护 2.0 的具体要求包括感知节点设备物理防护、接入控制、入侵防范、感知节点设备安全、网关节点设备安全、抗数据重放、数据融合处理及感知节点管理这 8 大类要求, 其中, 感知节点设备安全、网关节点设备安全等明确要求采用可信 3.0[30]架构构建主动防护体系.

- 对物联网及边缘计算安全风险及技术

在2019 年北美开源峰会上, 微软公司的 Tarditi 提出了高度安全的物联网设备的 7 个不同属性, 包括证书的身份认证、故障报告、OTA 安全性等几个关键属性, 报告重点强调可信根及可信基是整个物联网安全的基石. 北京工业大学的宁振虎[31]提出将可信计算技术与物联网感知层安全机制相结合, 研究了感知层可信免疫技术、面向感知层的物联网可信网络连接技术以及感知网行为可信度量技术和感知网可信证明技术, 从整体上保障物联网感知层的安全可信.

- 物联网及边缘计算具体实践

亚马逊采用通用的 HTTP & MQTT 接入方案, 增加 IoT 设备监测; 微软 Azure 支持安全标准设备标识组合引擎(DICE)和硬件安全模块(HSM); 阿里云结合 GloablePlatform TEE 思路, 提出了 Link TEE 方案. 在物联网终端可信方面, 在过去的 20 年中, 我们看到了各种硬件安全解决方案和实践趋势, 从可信平台模块 TPM、ARM 的 TrustZone 和物理不可克隆函数(PUF), 到最近的进展, 如 Intel 的软件保护扩展(SGX)和控制流强制技术(CET), 攻击变得越来越隐晦难防, 威胁的维度在不断增加.

从国内外研究现状来看, 从硬件机制方面, 可以较好地解决物联网终端及边缘网关的安全可信. 但从微软、AWS、阿里等厂商的解决方案来看, 基于硬件可信保障机制还在逐步完善, 特别是针对物联网终端的可信 3.0 标准的实现以及动态度量技术的研究和工程应用还处于起步阶段.

2 SG-Edge 总体框架

电力物联网不仅仅是网络基础设施, 也不仅仅是物联网技术的运用, 是综合应用信通新技术, 通过信息物理融合与新一代电力系统相互渗透和智能互动, 实现能源电力生产消费各环节人、机、物的实时在线连接与融合发展, 逐步形成支撑我国能源互联网运行的基础设施.

2.1 智慧物联体系框架

2019 年 10 月, 国网公司发布由江苏电力公司牵头, 南瑞、华为、阿里等多家单位参与制定的智慧物联体系框架, 具体如图 1 所示. 其中,

- 电力物联网平台主要由物联管理平台与能力开放中心两部分组成, 向下管理物联代理、业务终端和网络资源, 向上支撑各类电力业务, 对外提供业务能力封装和能力开放 API 接口. 物联管理平台支持物联, 实现对网络拓扑链路的连接建立、保持及配置、网络资源的虚拟化编排管理、终端等设备的状态监测、集中配置、远程升级等, 以及内外网用户的身份认证、权限管理等功能; 能力开放中心对内外部业务应用提供开发环境及 API 调用接口, 支持第三方能力集成, 提供可复用的业务基础服务, 支持业务的 APP 化和快速开发, 支持应用终端 APP 的开发、发布及信息推送等;
- 边缘计算框架在功能架构上分硬件层、操作系统层、基础功能层和边缘服务层. 其中, 硬件层包括设

备唯一标识、可信计算模块等功能; 操作系统层包括系统监测、安全接入、应用隔离、可信度量等功能; 基础功能层包括子设备接入、物模型管理、消息队列等功能; 边缘服务层包括流计算、规则引擎等功能, 并支撑资源、数据、智能、应用管理等的云边协同. 在这里, 边缘计算主要应用在物联代理中, 支撑电力物联网实现 APP 控制、边缘数据共享、边缘计算、云边协同、APP 开发等功能.

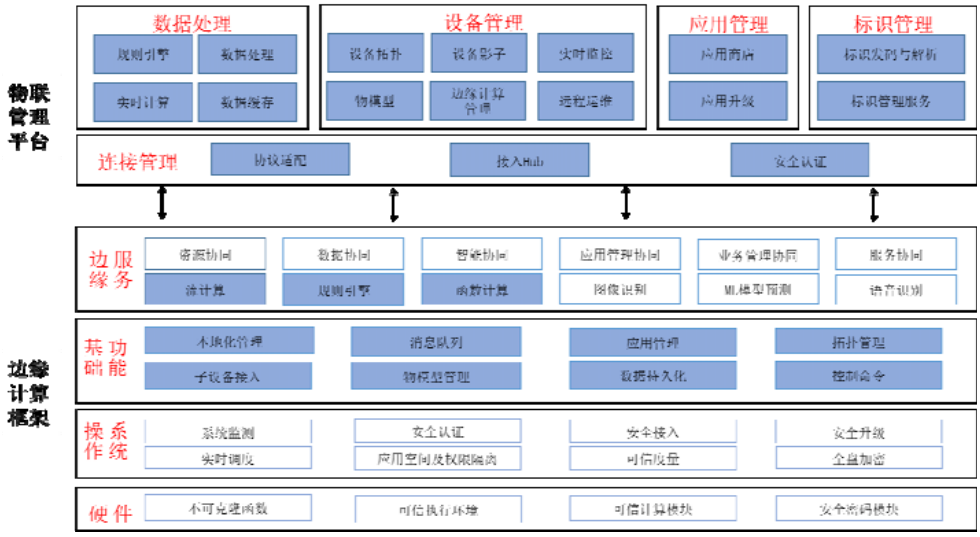


图 1　智慧物联体系框架

边缘计算框架对下衔接南向的子设备, 涉及各种异构的传感设备; 对上连接北向的物联管理平台, 支持云边协同. 总体而言, 存在如下 4 个方面关键需求.

- 关键性能
  - 首先, 电力物联网主要面向传统的电力工控实时系统, 系统具有明确的硬实时需求. 例如: 电力负荷控制系统属于硬实时系统, 需系统及时处理外部事件, 否则可能造成不可预期的后果;
  - 其次, 嵌入式实时系统存在严格的确定性需求. 系统的某些关键业务必须在确定的时间内完成;
- 安全可信: 开放的软件生态环境以及互联网技术引入到电力控制系统中, 也给现有封闭的嵌入式系统带来了安全隐患: 大规模的开放软件生态环境可能含有安全漏洞、未知后门; 高速的以太网接入方式也给黑客提供了便捷的攻击路径, 需要考虑到现有嵌入式系统的实时性、确定性要求; 同时, 频繁升级、打补丁会影响系统安全性, 需要引入安全可信的主动防御方法;
- 高可靠性: 部分嵌入式系统涉及到行业以及个人生命安全, 对于这类系统需要进行系统全面的失效分析, 从失效概率、危害大小、危害可控性等维度评估出业务模块的功能安全等级. 需要底层的边缘框架符合响应功能安全等级的要求, 并且提供故障监测控制、故障隔离以及故障恢复的功能;
- 智能生态: 首先, 智能物联网装备引入互联网、大数据、人工智能等新技术, 这些新技术需要边缘框架提供开放的智能软件生态, 新技术多数从 IT 行业发展而来, 依赖开放的智能软件生态; 其次, 智能物联网装备可能由多个系统相互协同完成工作任务, 因此需要边缘框架提供互联互通技术, 以及进一步提供互操作互调用的机制. 所以, 类似 iOS 或 Android 等优秀的系统都有自己特有的开发者框架, 一方面便于开发者使用, 另一方面, 将自己独有的 OS 设计理念融于到业务应用中.

2.2 可信边缘技术架构

在充分考虑开放性、可靠性、安全性及云边协同等需求的基础上, 我们融合 EdgeX、KubeEdge 与 OpenEdge

的设计理念，设计实现如图 2 所示的边缘计算框架 SG-Edge.

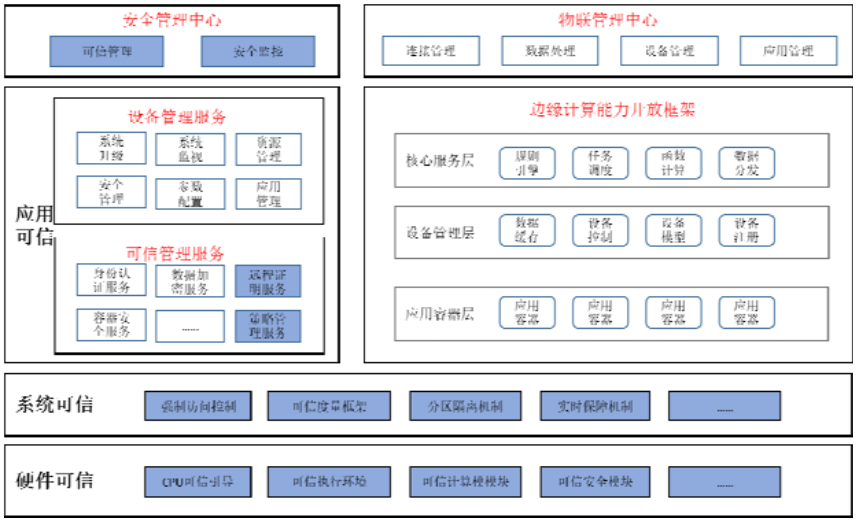


图 2 SG-Edge 可信边缘框架

整个框架在可信安全保障方面分为 4 个层面，包括硬件可信支撑层、操作系统可信增强层、可信服务层以及设备管理层. 4 层设计简述如下.

- 硬件支撑层. 包括基于密码技术的可信基以及基于硬件特性的可信启动和可信隔离技术;
- 系统可信层. 操作系统采用安全增强功能，具体包括以下安全功能: 三权分立、强制访问控制、MAC 强制访问控制、双因子认证、操作系统的双因子认证支持、磁盘加密、分区隔离以及实时支撑等功能;
- 可信服务层. 参考等级保护 2.0 要求，提供基于 TCM 的安全管理及可信度量、远程证明等服务;
- 设备管理层. 主要展开对边缘设备的边缘框架及边缘应用的安全管理，以及操作系统、固件等可信升级、边缘设备的安全监测等.

由于电力物联网涉及到关键电力设备控制功能，边缘设备作为边缘层的大脑，安全等级较一般边缘计算网关安全性有更高要求，如可信启动、安全升级、抗恶意攻击、安全监控，并符合等级保护三级以上安全要求. 结合国内外安全技术发展现状，特别是可信计算 3.0 主动防御的思想[30]，对电力物联网边缘网关安全防护提供了参考. 本文在安全操作系统增强基础之上[32]，重点开展以下两项安全可信保障机制研究: 首先，从 CPU 上电到 OS 内核加载过程的静态完整性，以及可信引导机制; 其次，研究操作系统运行过程中软件行为和业务行为完整性，即软件行为可信动态度量机制.

3 可信保障关键技术

边缘框架作为业务中间件，其安全性建立在可信体系和安全操作系统相关机制之上. 本文基于等级保护四级安全操作系统之上构建可信信任体系，重点关注可信引导和可信软件基的构建和保护，具体如下.

第 3.1.1 节分析利用新型硬件的可信引导机制 CRTM(如 SecureBoot, DICE 等)，实现边缘网关从 CPU 引导到 BootLoader、到操作系统内核可信验证，实现主动免疫，确保固件安全启动，安全可信升级. 由于各厂商所使用的硬件机制不一致，各厂商在 TCM 的基础上，采用厂商签名方法，并通过验证签名的方法来构建信任链; 同时，通过写 PCR 方式报告固件的版本，以便管理中心能够监测到固件的信息.

可信 3.0 中提及可信软件基和宿主操作系统是软件层两个并行的双栈，通过可信软件基的主动监控机制对操作系统中的监控点实施可信验证，可信软件基逻辑上和宿主操作系统是逻辑隔离，所以其安全性非常关

键. 如果可信软件基遭受攻击, 将导致整个可信体系失效. 为此, 第 3.1.2 节提出将可信软件基分成两个部分: 一部分作为主动监控代理, 一部分作为监控机制服务实施度量和判定. 监控机制服务采用类似 TEE 硬件机制进行保护机制或者定制化可信安全模块, 可进一步增强可信软件基的安全性.

第 3.2.1 节分析了当前软件度量的模型及方法, 在上下文关联模型及分析方法方面存在不足, 提出了基于知识图谱的软件行为建模方法和基于 CEP 的软件行为度量(以及攻击行为检测)的方法, 通过行为图谱的语义, 利用过滤、关联、聚合等技术, 持续地从行为流中度量软件行为模式和检测异常攻击行为.

由于软件动态度量机制无法 100%构建行为特征, 软件行为的模型需要动态学习和完善. 第 3.2.2 节结合动态完整性度量对象信息, 提出了“云端学习, 边缘决策”的动态行为度量框架. 同时, 结合 KillChain 模型[39]和 ATT&CK 知识库, 构建基于攻击视角的行为分析, 发现异常攻击. 态势感知与威胁监测作为可信保证体系的补充, 进一步提升物联网主动防御能力.

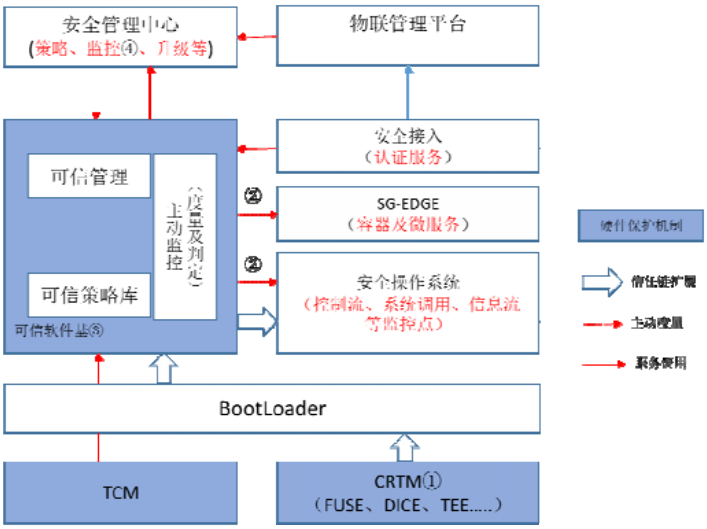


图 3 基于可信 3.0 可信技术体系

3.1 硬件可信启动及信任链构建

3.1.1 基于硬件可信技术设计思路

为了防范针对硬件平台固件的篡改行为, ARM、Intel、NXP 等厂商均提出了可信引导机制, 确保 CPU 能够从指定的程序进行引导, 从而为可信引导提供可信度量根核(CRTM). 另外, 为了保障移动终端的安全性, GlobalPlatform 提出了 GP TEE 安全方案. 随后, 一些厂商提出了 TrustZone[33]、SGX[34]等可信执行环境的方案. 另外, 考虑到 TPM 在物联网环境中应用存在功耗和成本的限制, TCG 也提出了替代 TPM 的 DICE (device identity composition engine)方案. 这些为物联网终端的安全所提供的应用保障, 可以很好地支撑边缘计算网关的可信引导和静态完整性度量以及可信软件的保护. 本文从可信启动、轻量可信计算、可信执行环境进行技术分析, 为第 3.1.2 节提供技术依据.

- 可信启动: Secure Boot 主要是采用 CPU SoC 机制, 系统软件采用签名认证的方式, 在设备出厂前对引导程序的镜像文件进行签名, 并将厂商公钥的 Hash 值写入芯片内部的一次性可编程区域(Fuse 熔断机制). 由于不同文件计算得到的 Hash 值不同, 采用 Secure Boot 方案的设备每次启动时都会先校验系统的 Hash 值, 即和芯片内的 Hash 值进行比较, 然后对镜像文件逐级校验, 实现从芯片到系统软件的链式校验过程. Secure Boot 首先解决了可信根问题, 其次, 通过厂商证书验证固件签名的方案实现了可信度量及验证, 从而仅需向 TPM 报告固件的版本信息, 其本身自带验证. 通过固件的层层验签确保了信任链构建;

- 轻量可信计算: TCG DIC 依靠简单的芯片功能和软件技术相结合, 提供强大密码的设备身份. 其核心思想是, 使用只有 DICE 可读的唯一设备密信(unique device secret, UDS)来创建每一层与硬件配置相关的密码. UDS 也可利用物理不可克隆函数 PUF (physical unclonable functions)技术来产生. 可信计算提供物理安全特征, 实现密钥安全存储、认证、信任根等功能. DICE 方案的核心是通过 HMAC (LayerN-key, H(LayerN+1 Code))方案扩展秘钥方式, 为各层提供密码保障. DICE 可以一直扩展到应用程序层, 在应用程序层中, 每个应用程序都获得一个唯一的别名密钥和证书, 用于对其标识进行编码. DICE 同样首先解决了可信根问题, 其次, 通过厂商证书验证固件签名的方案实现了可信度量及验证. 由于下层给上层提供了固件信息, 所以无需 TPM 即可报告固件信息. 通过固件的层层验签确保了信任链构建;
- 可信执行环境: TrustZone 作为 ARM 架构特有的硬件隔离机制, 基于 TrustZone 机制可以构建安全隔离环境、保护系统完整性、构建可信计算环境、构建安全服务和实现安全启动等. 具备 TrustZone 功能的 ARM CPU 可以通过 TEE 引导 REE, 提供 REE 类似 Secure Boot 的安全启动服务, TEE 可利用 TCM 或者其他安全密码模块为 REE 提供可信计算服务, 利用其特有的隔离特性, 可以为 REE 提供安全服务, 包括可信度量服务.

由于边缘代理和电力物联网设备对计算资源要求不同, 所以可分别采用不同的可信计算机制. 对于低处理能力的边缘计算网关或物联网设备, 可采用 DICE 或 Secure Boot 来链构建技术. 对于高安全等级的工控类边缘计算网关, 推荐采用 TrustZone 或者 TPM、TPCM 方案. 通过硬件可信引导和信任链构建, 满足等级保护 2.0 可信验证相关功能要求(“基于可信根对系统引导程序、系统程序、重要配置参数和通信应用程序等进行可信验证”). 硬件可信机制能够很好地满足静态可信验证, 但动态可信度量, 特别是基于硬件机制的可信软件基是目前研究的盲点. 为此, 本文提出了基于硬件安全特性的可信软件基保护方法.

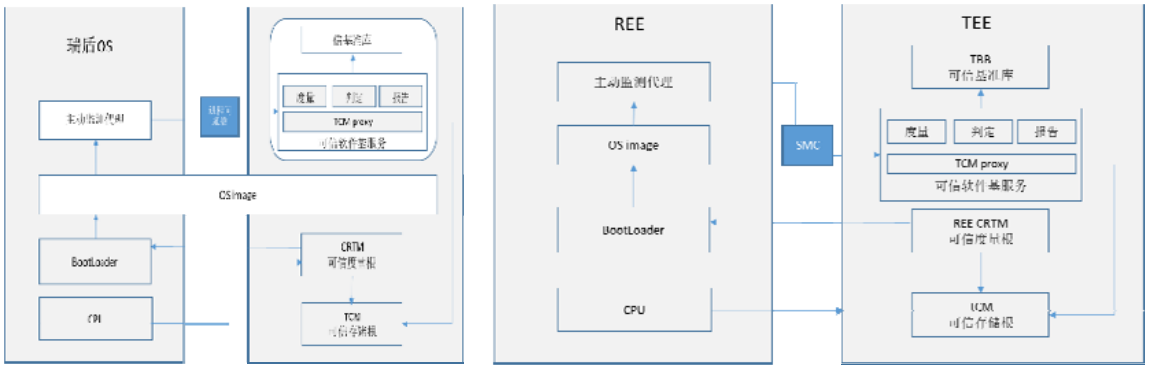
3.1.2 基于硬件安全特性的可信软件基保护方法研究

可信软件基(TSB)在可信计算体系中处于承上启下的核心地位, 对上保护宿主基础软件和应用的安全, 对下管理 TCM, 并承接信任链的传递. 可信软件基是对可信计算节点中的可信防护部件安全模型的一种抽象, 《GB/T 37935-2019》提供了可信软件基的安全要求, 从安全性角度来看, 可信软件基是可信体系的核心, 未见相关实现, 测试和验证标准尚不健全. 本文从保障操作系统安全性角度, 对 TrustZone 方案和可信安全模块(TSM)在可信软件基中的作用进行分析, 提出了符合边缘计算需要的高安全等级防护策略以及安全性和可靠性的评价方法(如图 4 所示).

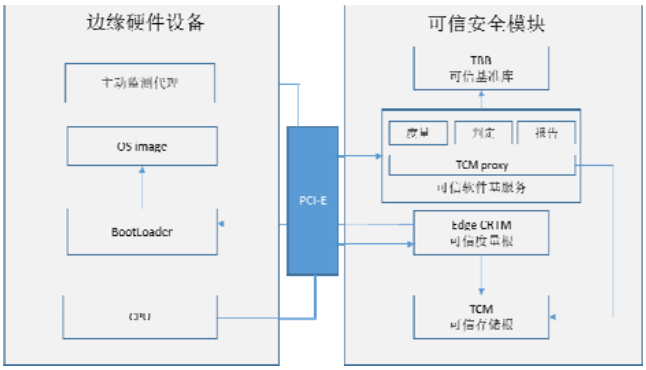
- 策略 1: 基于安全操作系统加固方案. 由于软件可信基和操作系统运行在同一内存空间, 其核心是确保安全可信基的启动时刻优先于操作系统其他模块. TSB 完全运行在 OS 空间, TSB 和 OS 内其他模块逻辑隔离, 性能较高, 但安全性较差, 特别是近年来发现的大部分攻击事件, 多基于操作系统层面;
- 策略 2: 基于 TrustZone 的硬件隔离方案, TEE 和 REE 通过调用 SMC 进行交互. 当不安全域的用户模式需要获取安全域的服务时, 首先需要进入到不安全域的特权模式, 在该模式下调用 SMC, 处理器将进入到 Monitor 模式, Monitor 模式备份不安全域的上下文, 然后进入到安全域的特权模式, 此时的运行环境是安全域的执行环境, 此后进入到安全域的用户模式, 执行相应的安全服务. 切换的过程仅仅是 CPU 运行地址空间的切换, 用时少速度快. 但对于复杂事件处理过程中需要记录事件序列, 并运行状态机, TEE 难以胜任, 因为 TEE 要应答每个 SMC 请求, 从效率上可能无法满足较为频繁的系统调用和 CFI 行为匹配;
- 策略 3: 基于 PCI-E 卡的 TSM 增强方案, 其隔离强度较 TEE 更高. 另外, TSM 可以更好地保护 TCM. 但一次 PCI-E 总线传输的时间较长, 频繁交互可能会使得系统运行速度下降.

为了进一步优化 TSM 性能, 采用集成 CPU 核、PCI-E 控制器和 FPGA 逻辑的 SOC 方案, 如 Zynq®-7000[35]. 其内部运行 Linux, 便于部署 CEP 引擎实现对行为流的处理, 满足基于时间序列的需求. 采用 FPGA 硬件逻辑实现正则引擎和状态机引擎, 可大大加快模式匹配速度. 为了提高 PCI-E 总线传输效率, 可以采用链式 DMA, 处理

需求以流的方式进入 TSM 卡, 处理结果以流的方式送出(如图 5 所示).



(a) 基于安全操作系统加固方案 (b) TrustZone 隔离方案



(c) 基于 PCI-E 卡的 TSM 增强方案

图 4 可信软件基保护方案

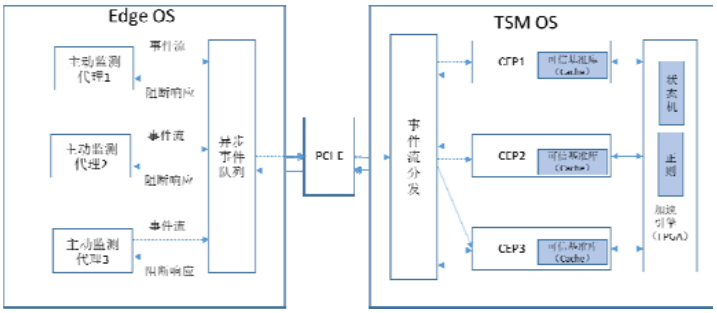


图 5 基于 CEP 的可信度量架构

TSM 方案基于以下几点考虑.

由于度量模块基于行为序列或状态机, 所以模式主动监测代理监测到的事件, 大部分无需逐个匹配, 无需逐个响应, 这样以事件流的方式缓冲在 PCI-E 驱动中,

(1) 采用 CEP 方案, 一方面可以满足时间序列的状态机模式匹配; 另一方面, 单一事件多路分发到不同的 CEP 引擎, 满足上下文关联需求, 具体关联模式分析详见第 3.2.1 节;

(2) 采用 CEP 方案, 与后面基于云端异常检测 CEP 方案对应, 这样可以实现云边协同. 云端的 CEP 事件处理规则和边缘网关 CEP 引擎实现原理一样, 简化了开发工作;

(3) 复杂的正则匹配和状态机匹配对 CPU 资源要求比较高, 所以采用 FPGA 相关 IP 或者自研 IP 能够大幅提升匹配性能;

(4) 框架基于异步调用的行为度量, 相对同步方式损失一定的安全性, 通过度量及判定的结果受到 TSM 安全保护, 通过远程证明方式及时报告安全管理中心进行决策.

基于 TSM 可信安全模块通过 PCI-E 总线与主机连接, 使用特定格式的报文与主机交互, 限制主机 CPU 对其读写访问的行为, 相比其他方式具有更高的安全性. TSM 作为整个系统的安全核心和可信起点, 通过可信引导、防物理攻击、密码保护和异常报告等功能, 实现整个系统的安全性设计要求.

(1) TSM 自带 NVME 或 SATA 固态硬盘, 强制 BIOS 或者 Bootloader 启动后从 TSM 引导, TSM 中的引导程序对其他硬件、固件和 OS 镜像进行检查验证, 确保其完整、合法;

(2) 为了防止物理攻击, 为 TSM 模块配置了金属保护罩, 内置电池和光电检测电路, 探测到物理攻击行为后, 内部电路将自动销毁关键数据;

(3) TSM 模块上电后进行自检, 运行过程中周期性自检, 对于软硬件的异常变化能够及时感知, 把异常及时向上报告, 并可根据异常的严重程度停止部分或者全部功能.

3.2 软件行为动态可信度量

3.2.1 动态可信度量设计思路

等级保护 2.0 明确提出, “在应用程序的所有执行环节对系统调用的主体、客体、操作可信验证, 并对中断、关键内存区域等执行资源进行可信验证, 并在检测到其可信性受到破坏时采取措施恢复. 同时, 将验证结果形成审计记录, 送到管理中心, 进行动态关联感知.”

软件行为学[41]将行为定义为主体使用函数(过程)对客体进行操作集合. 一般通过软件静态代码扫描或二进制分析抽取软件行为, 或者正常运行时收集运行过程信息建立行为模型, 然后根据待测行为与模型的偏离程度检测异常. 国内外学者在基于系统语义和行为可信度量方面开展了比较多的研究和探索, 主要包括 3 个方向.

(1) 控制流完整性, 其核心思想是限制程序运行中的控制转移, 使之始终处于原有的控制流图所限定的范围内. 具体做法是: 通过分析程序的控制流图, 获取间接转移指令(包括间接跳转、间接调用以及函数返回指令)目标的白名单, 并在运行过程中核对间接转移指令的目标是否在白名单中. 控制流劫持攻击往往会违背原有的控制流图, CFI 使得这种攻击行为难以实现, 从而保障软件系统的安全;

(2) 系统调用完整性, 其核心思想假设每一个程序的正常行为都可以由该程序正常运行时产生的系统调用序列来描述, 把这些正常的序列存放在库中, 称为正常库; 建立起正常库后, 监视该程序的运行状态, 当该程序产生的系统调用序列偏离了正常库时, 则可能产生了安全威胁;

(3) 信息流完整性, 其核心思想是: 通过分析多个不同安全类客体之间信息的流动关系, 确保不违反强制访问控制策略.

- 行为模型

常见的软件行为建模包括行为序列、行为树、有限状态机等表示, 这些模型表达方式基本以行为序列为主, 不能够完整表达行为的上下文语义. 很多研究和实践都基于行为上下文关联进行行为描述和检测. 如在控制流完整性方面, PathArmor[36]将 CFI 不变量和 CFG 中的控制流路径联系到一起, 运行时, 在执行路径上强制执行这些不变量. 在系统调用序列方面, SBO[42]基于对象的软件行为模型, 其状态由软件所关联的所有系统对象表示, 从而赋予状态的语义信息(如系统调用参数分析), 解决了不同行为与语义不相关问题. 在信息流完整性方面, PRIMA[43]通过关联 SELinux 的策略进行信息流完整性异常检测. 基于上下文关联的行为分析也可称为多维(多元)行为关联分析, 主要通过行为节点主客体及操作的属性(上下文约束)的关联分析, 其模型是多实体的语义关系网络, 可用知识图谱刻画和建模. 在行为知识图的基础语义主体、客体、操作的基础上, 通过丰富静态(主体、客体等属性)和动态(系统动态属性, 操作序列等)上下文信息, 然后通过基于图计算实现行为度量及异常检测. 基于知识图谱的软件行为建模能够统一表达控制流完整性、系统调用、信息流行为模型以及这 3 类行为模型的关联关系, 进一步丰富地表达复杂的行为模型.

- 行为建模

按照行为及其属性之间的关系, 又可以分为序列关系(如执行序列、时间序列)、空间分布关系(数据的多维空间分布)、图谱关系(数据表示成点, 通过边相连). 其中, 时间序列异常通过对时间分布上的数据建立时间序列模型(自回归滑动平均模型), 比如流量在时间上的分布, 用于发现终端数据在趋势上的不稳定因素; 行为序列异常需要终端数据中提取行为序列, 比如系统调用序列、命令执行序列, 之后建立正常序列模型(隐马尔可夫模型), 然后针对新的行为序列输入到模型中检测并发现异常的行为序列. 行为序列异常还可以通过 word2vec、tf-idf、n-gram 等自然语言处理算法将行为序列数据转化成多维特征向量, 然后使用聚类、孤立森林、高斯分布等多维异常检测算法发现行为序列在多维空间分布上的孤立点.

- 行为判定

对于安全领域异常检测, 高效的关联分析技术一般采用 CEP(复杂事件处理引擎)技术[38]. CEP 解决了对连续传入事件进行模式匹配的问题, 匹配的结果通常是从输入事件派生的复杂事件. CEP 引擎应用于无限数据流匹配, 对于输入的数据流立即进行处理, 一旦查询到与行为模式序列相匹配的事件, 结果就会立即发出. 因此, CEP 引擎具有较强的实时分析能力. CEP 引擎可以处理的事件关系包括: (1) 时间顺序关系; (2) 聚合关系; (3) 层次关系; (4) 依赖关系; (5) 因果关系. CEP 不仅可以度量软件行为, 同样也可以较方便地实现攻击检测. 常见的入侵检测等技术识别特定环境的恶意行为, 但缺乏系统性的攻击路径分析, 特别是对 APT 等攻击手段检测准确率低. 常见的攻击异常检测包括基于规则的异常检测、基于时间序列的异常检测、基于不变量的异常检测、离群点异常检测等. 其中, 基于规则和时间序列的异常检测方法可等同本文所述的软件行为动态度量.

基于以上分析, 本文提出了一种基于知识图谱的软件行为建模方法和基于 CEP 的软件行为度量(以及攻击行为检测)的方法, 通过行为图谱的语义, 利用过滤、关联、聚合等技术, 持续地从行为流中度量软件行为模式和检测异常攻击行为.

3.2.2 基于云边协同的可信判定技术研究

为了实现行为动态建模, 本文提出了基于云边协同的可信判定技术研究, 如图 6 所述: 边缘侧对软件行为可信行为和异常行为进行判定, 采用 CEP 方法可以实现行为序列及多层行为关联; 在云端, 基于知识图谱与外部威胁情报等信息, 利用 CEP 技术实施软件行为的学习及攻击链的异常检测等. 云端学习的可信行为模型及时通过安全可信管理下发到边缘侧. 边缘 CEP 引擎和云端 CEP 引擎采用相通的策略, 确保云边协同的一致性.

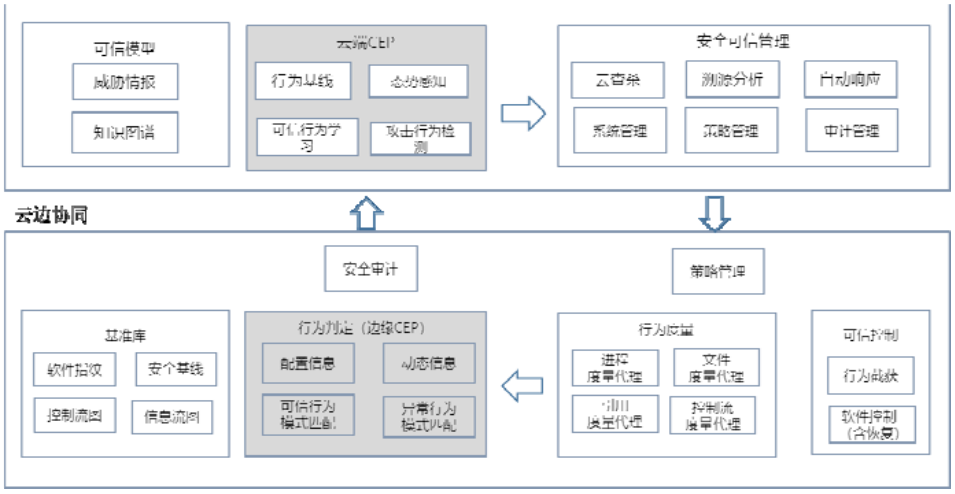


图 6 基于云边协同的可信判定架构

该框架基于边缘侧操作系统, 能够实施以下保障措施.

1) 操作系统采用强制访问控制框架, 如 SeLinux, 实施对访问控制策略的监控;

2) 操作系统在不干扰系统原有安全策略的基础上, 通过 LSM(Linux security module)技术在内核中增加一层访问控制策略, 实现对系统调用函数的监控;
3) 操作系统通过对进程关键函数进行 HOOK, 如 Auditd, 能够对进程内存空间进行监控, 从而获取用户进程空间控制流信息;
4) 操作系统运行的边缘程序在编译连接过程中已生成控制流 CFG, 并完成二进制相关插桩功能[44], 为控制流完整性和系统调用关联提供信息;
5) 通过对微应用中能够引发漏洞的关键函数采用 RSAP 技术[45]进行 HOOK, 实现对容器应用和微服务应用调用链以及访问行为的监控;
6) 边缘层通过 TSM 可信加密模块独立实现 CEP 引擎, 确保 CEP 行为判定不影响边缘 OS 性能.

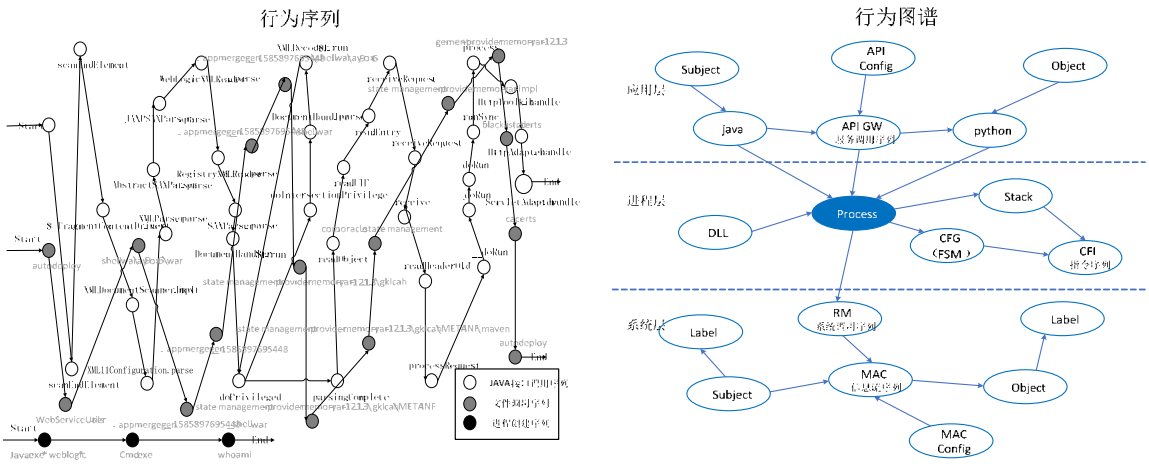
采集后的信息进一步归一化, 存入到行为(事件)集, 进入规则引擎进行判定. 因为从静态代码分析无法完全捕获软件行为, 本文采用自学习方式, 在云端对程序运行阶段采集的行为集开展行为学习, 生成行为图谱, 动态更新到边缘设备, 经过测试后确定为软件行为基. 以边缘计算框架核心程序 Agent 的加密 Webshell 监测为例, 通常, 加密变种的 Webshell 基于流量的方式比较难以监测, 加密 Webshell 植入到主机上执行的一系列行为动作可识别为异常. 图 7 详细描述了基于图谱和规则引擎的软件行为建模及异常检测.

1) 事件归一化: 事件归一化基于实时计算技术, 输入多源异构行为事件(控制流信息、接口调用序列、系统调用序列、命令执行序列、进程创建序列、文件读写序列等); 定义统一的事件模型, 包括事件时间、事件类型、事件来源、设备编号、事件内容等; 通过对输入的事件流进行解析、过滤、降噪、聚合、去重、关联、增强, 生成归一化的事件流; 然后, 将事件流推送到规则引擎中进行行为判定. 如图 7(a)所示为“冰蝎”Webshell 在执行命令时, 监测到的接口调用序列、文件读写序列和进程创建序列;
2) 行为图谱: 行为图谱的构建首先要确定图谱中的实体与关系, 其中,
   (1) 实体包括设备编号、软件、文件、进程、服务、系统调用、命令等实体信息;
   (2) 关系通常分为直接关系和间接关系, 其中, 直接关系是指从行为事件可以直接得到的关系, 比如设备和软件的安装关系、服务和进程的映射关系、文件和命令的执行关系、进程和服务的请求关系等; 对于间接关系, 则需要通过复杂的数据挖掘来获得, 如两个进程在一段时间内的系统调用序列, 通过将序列进行向量化, 然后进行相似性比较, 如得出两个进程的系统调用序列具有相似性, 建立进程之间的相似性关系;
   (3) 确定了实体与关系后, 按照实体与实体的关系对归一化事件流构建行为图谱, 并存储到图数据库中;
   (4) 对于构建好的行为图谱, 采用图推理算法(图嵌入技术、低维向量表示和路径排序算法等)进行关系推演, 如路径排序算法通过将两个实体之间的路径作为特征来预测两个实体之间的隐藏关系;

   图 7(b)是以进程为对象构建的图谱关系图, 能够在进程外部应用层父子进程关系、外部 API 调用关系、资源对象访问关系进行关联; 在进程内部的依赖库、堆栈信息和指令执行流进行关联; 在进程操作系统层面, 将进程所具备的安全属性、系统调用功能号等内核对象进行关联;
3) 行为判定: 基于行为图谱构建异常行为检测模型, 通过构建正常行为模型推导异常行为. 一是基于行为规则进行判断, 通过构建异常行为规则库, 命中行为规则库则判定为异常; 二是基于行为相似度计算进行判断, 计算离群点, 针对偏离正常行为序列的特征判断为异常. 如图 7(c)所示, 将监测到的文件读写序列规则、进程行为序列规则和接口调用序列, 以进程为对象进行聚合关联, 将不能匹配正常行为规则的序列, 判断为异常.

针对电力物联网的攻击一般带有明确的攻击意图, 如攻击者一般采用 Kill Chain 模型[43], 分为侦察、武器化、散布、恶用、设置、命令与控制、目标达成这 7 个阶段. 另外, MITRE 基于 KillChain 的模型, 提出了 ATT&CK (adversarial tactics, techniques, and common knowledge)[44], 将已知攻击者行为转换为结构化列表, 将这些已知的行为汇总成战术和技术, 并通过几个矩阵以及结构化威胁信息表达式(structured threat information expression,

STIX)、指标信息的可信自动化交换(trusted automated exchange of indicator information, TAXII)来表示. 由于此列表相当全面地呈现了攻击者在攻击网络时所采用的行为, 因此对于各种进攻性和防御性度量、表示和其他机制都非常有用. 为此, 本文提出在单一软件行为度量基础上, 在云端实现高级威胁识别, 作为可信体系的补充.



(a) 行为序列　　(b) 行为图谱

行为规则匹配

```
文件行为规则1：p1, [%MonitorDir%]  read|write [%FileName%]
                as EventSeq1
进程行为规则2：p2, [%ProcessName%] create [%SubProcessName%]
                as EventSeq2
Java接口调用规则3：p3, In Stack, [%FuncName%] call [%FuncName%]
                as EventSeq3

              With EventSeq1* | EventSeq2* | EventSeq3*
              Match  [NormEventSeq]
              Return  Norm|Anorm
```

(c) 行为规则

图 7　基于知识图谱的软件行为建模及行为判定

结合知识图谱的环境信息, 并结合终端检测响应(endpoint detection & response, EDR)技术实现边缘网关威胁发现, 在安全管理中心实现电力物联网态势感知, 主要功能包括:

(1) 风险识别. 结合安全基线标准, 通过账户、网络、进程、系统配置等多维度风险检测, 发现不符合安全管理规范的边缘网关;

(2) 攻击检测. 本文前面所述的行为度量, 结合威胁情报以及相关安全规则, 对端口扫描攻击、暴力破解攻击、恶意脚本攻击、系统漏洞攻击、Webshell 攻击等可疑行为进行更高层次的异常行为检测;

(3) 自动响应. 例如基于边缘网关的容器的微隔离, 控制边缘网关的应用的流量行为, 实现对异常 APP 的隔离;

(4) 态势感知. 通过综合分析边缘网关及其相关联的网络、物联网应用的异常行为, 实时感知和预测物联网的安全态势.

3.3 可靠性保障措施

3.3.1 业务故障隔离技术

在高可靠领域, 故障隔离是解决功能安全的核心关键. SG-Edge 采用微服务架构设计, 对于用户态的不同

功能安全等级业务提供分区隔离机制. 当某个业务发生故障并失效后, 其他分区业务不受影响. 具体技术包括分区间的时间隔离机制、空间隔离机制、权限隔离、硬件隔离机制, 其中,

- 空间隔离是最为直接的权域控制手段, 进程主要是内存地址空间的隔离, 引入容器后, 实现了文件系统、网络、进程 PID、用户等的隔离, 其中, Namespace 用于空间隔离, Cgroup 则用于资源隔离;
- 在权限隔离方面:
  - 一方面, 借鉴 Linux 沙箱机制, 应用程序之间无法交互, 运行在进程沙箱内的应用程序没有被分配权限, 无法访问系统或资源. SG-Edge 应用程序的“沙箱”机制确保互相不具备信任关系的应用程序相互隔离, 独自运行;
  - 另一方面, 基于 OS 层面的强制访问控制, 实现业务权域最小化运行, 主要通过访问控制模型来限定业务的权限访问范围.

  为了丰富 SG-Edge 的权限控制机制, 我们引入了 CAP、BLP、BIBA 等强制访问控制模式: CAP 模型只赋予用户以及每个特权进程能够完成其功能的最小能力, 实现业务最小运行权域, 包括允许访问网络、访问外设、关机等 30 多项权限等; BLP 和 BIBA 是传统机密性和完整性模型在系统中的实现.

3.3.2 基于边边协同的任务保障机制

由于边缘网关资源受限, 导致计算与传输能力的不稳定. 造成该不稳定的因素十分复杂且难以避免, 包括间歇性的组件故障、底层资源争用、广域网络拥塞等, 这些不确定因素会导致各边缘的实际服务能力在短时间内发生较大波动, 从而使其承载计算任务的真实执行时间可能会超出预期, 成为“慢”任务. 尤其是在部分负载较高的“热点”边缘, 这样的不稳定性会加剧, 甚至部分边缘会出现故障而不可达. “慢”任务不仅直接造成其响应时间增长, 也可能导致其所在的整个应用或作业无法完成.

为可能变“慢”的任务复制多个任务副本运行于其他负载较低的边缘, 这些副本中最快完成的即能返回结果或推进后续计算. 通过这样的冗余机制, 能够有效保障任务在预期时间完成, 但过多的任务副本也会增加系统的整体负载. 此外, 由于不同边缘适合运行不同类型的任务, 并且任务副本的运行也需要边缘间数据转移, 因此, 如何选择合适的边缘部署任务副本也是难点问题. 为此, SG-Edge 通过感知各个异构边缘任务执行的时延分布状况, 综合考虑任务特点与系统整体负载, 建立了在线任务复制机制, 为各个任务在线确定副本数量与最佳复制位置, 在不稳定的边缘环境中, 保障整体系统的执行效率.

SG-Edge 基于各边缘的资源管理器反馈的本地任务执行情况, 为各边缘建立一个独立的延迟因子分布模型. 当移动用户将任务请求提交至一个边缘, 该边缘会安排任务执行, 并将任务提交至云端, 由 SG-Edge 根据任务特征、所在边缘与系统整体负载, 通过任务副本预估机制与任务副本部署机制, 确定任务副本数量并在多个边缘并行执行, 在获得最快响应结果后, 即返回结果并中止其他任务副本的执行(如图 8 所示).

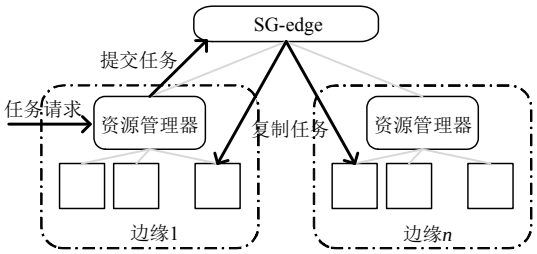


图 8 边边协同任务保障机制

任务副本预估机制基于 SRPT(shortest remaining processing time, 最短剩余处理时间)原则和公平共享原则为各任务预估所需副本数. SG-Edge 按照剩余时间短任务优先原则为任务排序, 并选择前一部分任务公平共享整个系统的计算节点, 设定选择比例为ε, 剩余 1–ε比例的任务等待在后续调度周期分配资源. 该比例值的最佳设定在不同系统环境中是变化的, 需凭借执行经验获得. 如果一个任务公平共享得到的节点数多于 1, 则额外的节点可用于执行副本, 并将额外节点数作为该任务的预估副本数.

任务副本部署机制基于各边缘的资源性能模型, 将各边缘的空闲节点分配给任务运行预估副本. SG-Edge 迭代地为任务分配副本执行节点, 即: 在每一轮中, 依次为任务分配一个空闲节点, 该节点属于当前期望资源性能最优的边缘. 当一个任务的预估副本已全部拥有执行节点时, 停止为任务分配额外节点. 当所有任务都无需额外节点时, 停止迭代. 采用迭代式分配是因为对于一个任务而言, 开启副本的边际收益会随着副本数的增加急剧减少. 因此, 以更少的副本为高优先级任务提供服务不会有明显的性能损失, 而节省的节点可以分配给那些还没有副本的低优先级任务, 从而在平均性能上得到显著提升.

进一步分析发现: 在任务副本部署中, 不必为任务所有的预估副本分配执行节点, 只要任务的预期执行时间达到了某一阈值, 即可停止为其分配额外的节点. 这样可节省部分资源, 以备系统中即将到达任务的资源需求. 该阈值设定为任务的固有执行时长乘以$(1+\varepsilon)u(t)$, 其中, $u(t)$为当前系统的节点利用率. 这意味着利用率越高, 任务的预期执行时延阈值越高, 则满足该阈值所需的副本数就越少. 这是符合预期的, 当系统负载重时, 应适当减少副本使用, 以保障系统中常规任务的执行效率.

3.3.3 实时可靠性保障方法

部分电力业务有一定的实时性要求, 这在控制业务中尤为显著. 最大中断响应时间是反映系统实时性的最重要指标, 它表征了一个业务中断任务可能的最长等待时间. 一般的 Linux 类操作系统的最大终端响应时间往往在 200 μs 左右, 难以满足实时性的要求. SG-Edge 基于瑞盾安全操作系统[37]构建, 瑞盾安全操作系统在内核层之下单独构建了一层“微内核 OS”, 用于分发中断任务, 上面同时运行瑞盾 Linux 内核以及瑞盾实时内核两个域, 分别用来支持传统应用和实时业务, 这样可以确保实时任务一定能够得到实时响应. 实验结果表明: SG-Edge 最大中断响应时间为 10 μs 左右(与 VxWorks 等实时操作系统相当), 可有效支撑现有的电力物联网强实时业务(如图 9 所示).

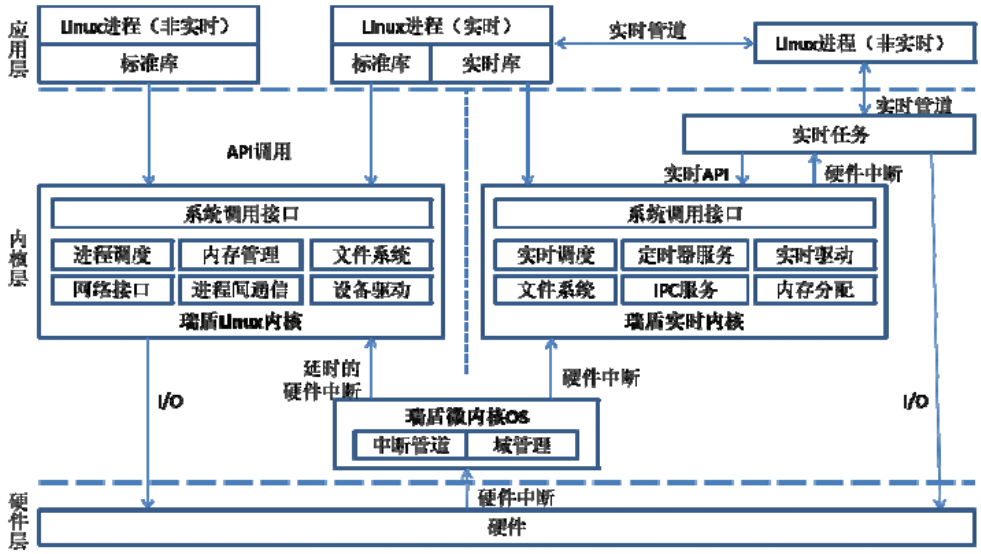


图 9 硬实时支持架构

4 实验与分析

为了验证 SG-Edge 框架的有效性和安全性, 本节将从功能、性能等方面对其开展实验验证工作. 其中, 因为边缘框架与物联管理平台的云边交互协议适合由平台侧发起测试, 所以部分测试从物联管理平台侧发起, 以此验证边缘物联代理中的边缘计算框架的支持情况. 本次测试的对象包括 SG-Edge、OpenEdge、Edgex、KubeEdge 这 4 个主流边缘计算框架.

4.1 功能测试

针对电网侧和用户侧实际业务需求, 我们在江苏电力有限公司溧水选取综合能源服务示范点开展了应用场景验证测试, 重点验证边缘物联框架对各个应用场景的支撑能力. 开展从终端、物联代理、物联管理平台到业务应用的端到端业务验证, 测试边缘框架南向、北向及完整链路下的联通性. 每个业务场景都基于完全相同的南向、北向环境, 完成各边缘框架的接入. 主要开展台区、配电站房、输电线路、综合能源服务等应用场景验证. 以台区场景为例说明验证内容. 为了提升台区配电网运行管理与用电客户服务水平, 实现台区侧、低压线路侧、用户侧等感知信息自动采集, 通过部署环境传感器、监测单元、智能电表等端设备实现台区状态的感知, 由智能配变终端 TTU、集中器等边设备实现信息汇集处理, 采用无线专网/公网等通信方式将采集数据传送至内网系统(如图 10 所示).

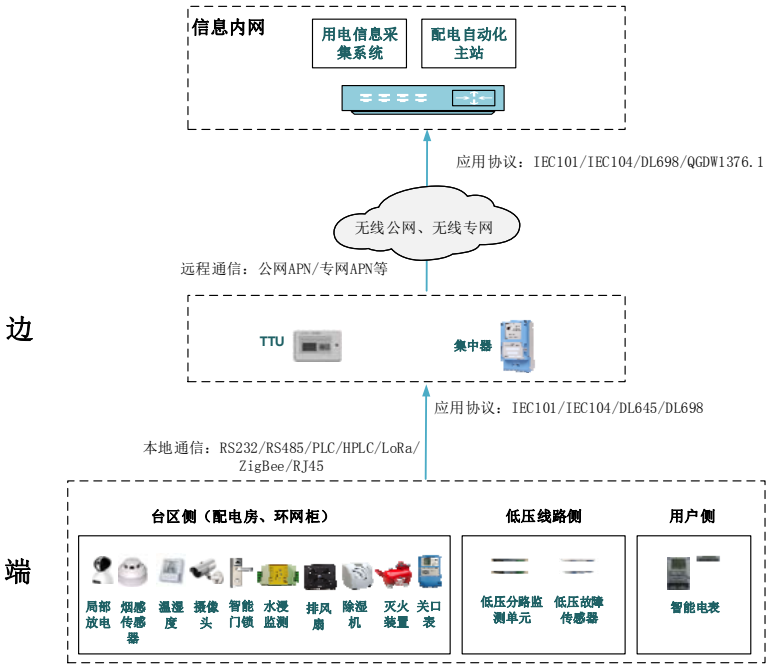


图 10 台区采集概述

本项目测试主要通过低压台区场景, 对电表数据上报、电表数据召测、电表控制指令下发、配电监测数据上报、断路器远程调试等电力物联网业务场景进行功能性及非功能性测试验证, 验证边缘物联框架对电力物联网业务的适配度.

对边缘框架的基础功能进行验证, 主要包括设备接入能力、设备管理能力、模型定义及下发、设备影子、数据收集、数据分发、固件升级、APP 管理、数据通信能力、规则引擎、运维管理这 10 项功能.

SG-Edge、OpenEdge、Edgex、KubeEdge 总体测试结果见表 1, 其中, ●表示完全支持, ○表示不支持, ◎表示部分支持.

从测试结果可以看出: SG-Edge 通过了所有功能、非功能、性能以及应用场景的测试, 表现最为完备; 而 Edgex 在安全性以及非功能性测试方面表现得不是很理想; KubeEdge 在非功能性测试和应用场景方面表现得不理想; OpenEdge 则在部分应用场景中表现得不是很好, 但其他方面表现相对优异. 由此可见: 从功能性、非功能性、性能以及应用场景等各个方面综合考虑, SG-Edge 目前是最为适应于电力物联网的边缘计算框架.

表 1 测试结果

验证内容	测试项	测试点	OpenEdge	Edgex	KubeEdge	SG-Edge
功能方面	设备接入	设备注册及接入	●	◎	●	●
		提供接入 SDK	●	◎	○	●
		数据传输协议	●	◎	●	●
		租户与物联代理关系	●	◎	●	●
	模型定义及下发	物模型定义	●	◎	●	●
		物模型下发	●	◎	●	●
		根据模型上报数据	●	●	●	●
	设备影子	设备影子编辑	●	●	◎	●
		数据上报	●	◎	◎	●
		状态变更	●	●	◎	●
	设备通信能力	MQTT 协议	●	●	●	●
		离线存储能力	●	●	●	●
	数据收集	设备消息导入消息队列	●	●	●	●
		数据临时存储	●	●	◎	●
		数据错误重传	●	●	◎	●
	数据分发	数据分发	●	●	●	●
		数据订阅	●	●	●	●
		下发数据指令	●	●	●	●
	固件升级	固件升级	◎	●	◎	●
	APP 管理	对 APP 上下架管理	●	●	◎	●
		对 APP 的远程升级	●	●	◎	●
		APP 远程配置	●	●	◎	●
		APP 版本管理	●	●	◎	●
	规则引擎	规则配置	●	●	◎	●
		类 SQL 语法和基础语义操作	●	●	◎	●
	运维管理	运维管理	◎	◎	●	●
非功能方面	远程配置	设备远程配置	●	○	○	●
		操作系统远程配置	●	●	○	●
	远程监测	操作系统远程监测	●	●	○	●
		终端运行状态监测	●	●	○	●
		应用运行状态监测	●	●	○	●
		告警信息管理	●	●	○	●
	远程调试	边缘代理设备远程调试	●	●	○	●
		终端设备进行远程调试	●	●	○	●
	可靠性	离线设备批量上线成功率	●	●	○	●
		集群高可用部署	●	●	○	●
	安全性	边缘代理设备接入安全	●	●	○	●
		边缘代理设备传输安全	●	●	●	●
		消息发布订阅安全	●	○	●	●
		API 鉴权	●	○	●	●
		平台登录安全	●	○	●	●
		平台使用安全	●	○	●	●
	灵活性	南向接口	●	○	●	●
		北向接口	●	○	●	●
	开放性	北向开放性	●	○	●	●
		南向开放性	●	○	●	●
	松耦合性	支持主流数据库	◎	○	○	●
		支持灰度更新能力	●	○	○	●
		支持组件自动扩缩	●	○	○	●

表 1 测试结果(续)

验证内容	测试项	测试点	OpenEdge	Edgex	KubeEdge	SG-Edge
性能方面	性能方面	最大同时连接数	◎	○	○	●
		每秒能处理的消息数	●	○	○	●
		下发指令的性能	●	○	○	●
		最大在线用户数	●	○	○	●
		核心功能平均响应时间	◎	○	○	●
		支持 8 小时持续上报信息	◎	○	○	●
		支持 8 小时持续下发指令	◎	○	○	●
应用场景	台区场景	电表数据上报	◎	●	●	●
		召测电表数据	○	○	○	●
		遥控电表状态	○	○	○	●
		配电数据上报	●	●	●	●
		断路器远程调试	●	●	●	●

4.2 安全性测试

为了验证 SG-Edge 的安全性, 我们选取 OWASP 发布的 IoT DRAFT[40]项目进行实验验证. 对照安全攻击面进行渗透测试及对比验证测试, SG-Edge 在通用安全、Web 接口、认证鉴权等方面均有相应的保护机制(见表 2).

表 2 OWASP IoT 攻击面

攻击面	安全隐患	测试结果
通用	互通性标准、数据治理、系统范围的失效、组件间隐式信任、注册安全、旧系统、缺失访问程序	已保护
设备内存	敏感数据、用户名明文、密码明文、第三方凭据、加密密钥	已保护
设备物联接口	固件解压、用户命令行、管理命令行、特权滥用、重置至不安全状态、移除存储设备、抗干扰、调试端口、UART、JTAG/SWD、设备 ID/串口端口暴露	已保护
设备 Web 接口	标准的 Web 应用程序漏洞、OWASP Top10、OWASP ASVS、OWASP Testing Guide、凭据管理漏洞、用户名枚举、弱密码、账户锁定、已知默认凭据、不安全的密码恢复机制	已保护
设备固件	敏感数据暴露、后门账号、硬编码凭据、加密秘钥、加密(对称、非对称)、敏感信息、敏感 URL 暴露、固件版本显示/最新更新日期、漏洞服务(Web, SSH, TFtp 等)、验证旧的软件版本及可能的攻击(健康监控、心跳、SSH、旧的 Php 版本等等)、安全相关 API 暴露、固件下载的可能性	已保护
设备网络服务	信息暴露、用户命令行、管理员命令行、注入、拒绝服务、未加密的服务、糟糕的加密、测试/开发服务、缓存溢出、UPnP、可攻击的 UDP 服务、Dos、设备固件 OTA 更新阻塞、固件传输使用了不安全的渠道(例如为使用 TLS)、Replay 攻击、缺乏有效载荷验证、缺乏消息完整性验证、凭据管理漏洞、用户名枚举、弱密码、账户锁定、已知默认凭据、不安全的密码恢复机制	已保护
管理员功能	标准的 Web 应用程序漏洞、OWASP Top10、OWASP ASVS、OWASP Testing 凭据管理漏洞、用户名枚举、弱密码、账户锁定、已知默认凭据、不安全的密码恢复机制、安全/加密选项、日志选项、双重认证机制、检查不安全的直接引用对象、无法擦除的设备	已保护
本地数据存储	数据未加密、使用已知密钥加密数据、缺乏数据完整性检查、使用静态相同的加密/解密密钥	已保护
第三方后端 API	发送未加密的个人信息、加密发送个人信息、设备信息泄露、位置泄露	已保护
更新机制	更新传输数据未加密、更新无签名、更新校验、更新验证、更新区可写、恶意更新、缺乏更新机制、没有手工更新机制	已保护
网络流量	局域网、局域网到 Internet、短距离传输、非标准网络、无线(WIFI, Z-wave, XBee, Zigbee, Bluetooth, LoRA)协议模糊测试	已保护
认证/授权	认证/授权相关数据暴露(例如 session key, token, cookie 等)重用、设备到设备身份验证、设备到移动应用身份验证、设备到云服务身份验证、移动应用到云服务身份验证、Web 应用到云服务身份验证、缺乏动态身份验证	已保护
隐私	用户数据暴露、用户/设备位置暴露	已保护

4.3 性能测试

4.3.1 可信性能测试

本方案中, 为评估可信度量对系统及应用带来的性能损失, 我们通过两个实验进行性能测试: 首先, 抛开框架本身, 纯粹地进行可信机制测试; 其次是业务影响测试, 本实验基于 AB 工具来探测可信机制本身对 SG-Edge 框架承载的 Web 业务的影响. 两项测试结果表明: 安全机制无论对框架还是对系统本身, 其性能影响均小于 5%.

为了评估不同可信软件基保护方法对系统性能的影响, 我们设计了以下实验: 将可信软件基软件分别部署在操作系统、TrustZone (TEE)和可信安全模块(TSM)中, 针对不同事件类型, 如时间顺序关系、聚合关系、层次关系、依赖关系、因果关系, 以 CEP 引擎将上述类型事件的行为判定时间、整体响应时间作为衡量指标, 评价不同保护方案的性能. 边缘网关的硬件配置如下: 主频 1 GHz 内存 1 G, 使用瑞盾操作系统, SG-Edge 框架. 可信安全模块 TSM 的配置如下: ZYNQ-7000 SOC, ARM Cortex-A9 双核, 主频 800 MHz, 内存 1 G, PCI-E X4; TEE 使用 ARM Cortex-A53, 主频 1 GHz, 软件使用 OpenTEE. 测试结果表明: TSM、TEE 与操作系统在行为判定时性能类似, 但整体响应时间操作系统优于 TEE 和 TSM. TSB 性能测试可见表 3.

表 3 TSB 性能测试

类型	OS			TEE		TSM	
	度量采集耗时(ms)	行为判定时间(ns/op)	整体响应时间(ms)	行为判定时间(ns/op)	整体响应时间(ms)	行为判定时间(ns/op)	整体响应时间(ms)
时间顺序关系	21.52	3 398	22.34	3 415	23.05	3 802	28.90
聚合关系	20.82	3 132	21.36	3 203	21.96	3 704	26.98
层次关系	21.09	1 518	21.83	1 602	22.09	2 307	27.23
依赖关系	22.53	2 625	23.29	2 663	22.89	3 105	27.50
因果关系	22.76	1 077	23.54	1 104	24.15	1 749	29.01

AB 的全称是 ApacheBench, 是 Apache 附带的专门用于 HTTP Server 的 BenchMark 测试, 可以同时模拟多个并发请求. AB 的原理是: 创建多个并发访问线程, 同时对某一 URL 地址进行访问. 在固定总量的请求次数前提下, 我们通过改变每次请求的并发次数, 观察每次请求响应时间, 以此来说明在安全增强的情况下, 系统 Apache 处理能力的变化情况. 通过 AB 工具的测试数据显示(如图 11 所示), 在安全增强的情况下, Apache 处理请求的响应时长略有增加, 平均大概延迟了 5%左右, 说明可信机制带来的性能损失在可接受的范围内.

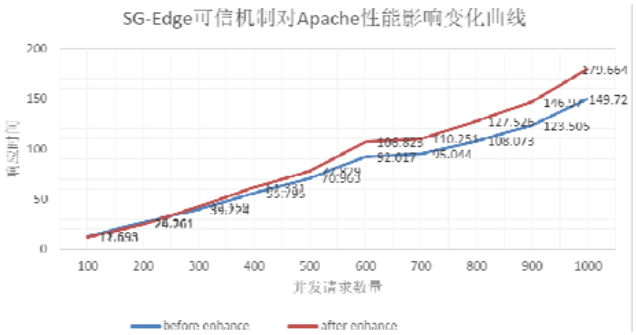


图 11 性能对比

4.3.2 边边协同性能

为评估该在线任务保障机制的性能, 我们进行了以下实验, 将当前常用的两类复制机制与 SG-Edge 的任务保障机制进行了对比, 以平均任务完成时间作为衡量指标. 这两类任务机制分别是基于监控的被动式复制和启发式克隆: 前者监控任务运行并收集执行信息, 在发现“慢”任务后, 为其启动任务副本以减少异常任务的影响; 后者与我们的机制类似, 在任务执行时就并行执行多个副本, 但副本数量仅有单一任务的状态信息确定副本量, 并未考虑整体负载与多个任务间的优先关系.

实验设定了 100 个边缘, 每个边缘平均计算能力为并行执行 30 个任务, 基于偏斜参数为 2 的 Zipf 分布边缘计算能力的分布. 任务负载参考 Facebook 商用 Hadoop 集群中的任务, 并基于泊松过程设定任务提交间隔, 构建不同的系统负载情况, 泊松参数值越高, 任务提交间隔越短, 则系统负载越重. 实验中设定轻负载的泊松参数值为 0.02, 中等负载为 0.07, 高负载为 0.15. 图 12 展示: 无论在哪种负载下, SG-Edge 的任务保障机制都使得任务的平均完成时间最短.

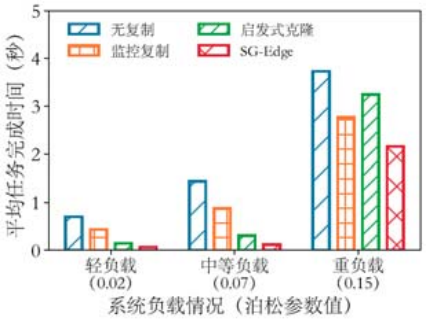


图 12 平均任务完成时间

5 结 论

边缘计算是指在靠近物或数据源头的一侧, 就近提供最近端服务. 其应用程序在边缘侧发起, 产生更快的网络服务响应, 满足行业在实时业务、应用智能、安全与隐私保护等方面的基本需求. 边缘计算处于物理实体和工业连接之间, 或处于物理实体的顶端. 在电力物联网的建设中, 构建一种适应电力物联网的边缘计算框架是其实现物联体系的核心. 为此, 本文设计实现了一套可信边缘计算框架 SG-Edge, 重点在性能、安全可信、高可靠性等关键技术方面进行了研究. 最后, 通过其在功能性、非功能性、性能以及应用场景等方面, 与 OpenEdge、Edgex、KubeEdge 等框架的对比测试, 说明了 SG-Edge 在业务满足性和适应性上的优势.

本文所述电力可信边缘物框架参考可信 3.0 架构要求, 结合操作系统安全加固和可信安全模块方式, 保障了可信体系静态度量方案的落地. 本文所述的基于云边协同的可信判定框架可进一步提升软件行为动态度量的准确性和及时性, 对提升电力物联体系态势感知提供技术支撑. SG-Edge 也可为云计算、工控系统等等级保护 2.0 提供技术参考. 从技术现状来看, 在静态度量方面, 第三信任链中的软件更新(BIOS 升级、OS 打补丁)信任需重新计算, 特别是热补丁或者系统运行后的程序升级, 还有待研究. 另外, 还存在 CRTM 存储在 TPM 之外, 容易受到攻击等难题. 另外, 在技术上准确提取软件的预期行为和获取软件的实际行为都还存在一定的困难, 包括行为的配置、行为模式的判定等.

未来, 我们将继续围绕现有工作, 不断优化 SG-Edge 的架构, 探索相关可信保障技术, 积极与用户、边缘设备及边缘应用厂商一道不断丰富电力物联网的应用生态, 助力电力物联网及国家工业互联网的建设.

References:

[1] Cao ZH, Lu YC, Lai SS, Yu ZY, Ma Y, Wang T. Research progress of sensor cloud based on edge computing. Ruan Jian Xue Bao/Journal of Software, 2019, 30(Suppl.(11)): 40−50 (in Chinese with English abstract). http.//www.jos.org.cn/1000-9825/19005.htm

[2] Denn WJ. Kill chain: The rise of the high-tech assassins. Military Review, 2016, 96(3).

[3] Shi W, Cao J, Zhang Q, *et al*. Edge computing: Vision and challenges. IEEE Internet of Things Journal, 2016, 3(5): 637−646.

[4] Beck MT, Werner M, Feld S, *et al*. Mobile edge computing: A taxonomy. In: Proc. of the Accepted for the 6th Int'l Conf. on Advances in Future Internet. 2014.

[5] Satyanarayanan M. The emergence of edge computing. Computer, 2017, 50(1): 30−39.

[6] Maheshwari S, Raychaudhuri D, Seskar I, *et al*. Scalability and performance evaluation of edge cloud systems for latency constrained applications. In: Proc. of the IEEE/ACM Symp. on Edge Computing. IEEE Computer Society, 2018.

[7] Ahmed R, Zaheer Z, Li R, *et al*. Harpocrates: Giving out your secrets and keeping them too. In: Proc. of the IEEE/ACM Symp. on Edge Computing. ACM, 2018.

[8] Chao M, Yang C, Zeng Y, *et al*. F-MStorm: Feedback-based online distributed mobile stream processing. In: Proc. of the IEEE/ACM Symp. on Edge Computing. ACM, 2018.

[9] Lohan V, Singh RP. Home automation using Internet of things. In: Proc. of the Advances in Data and Information Sciences. 2019.

[10] Ai Y, Peng M, Zhang K. Edge computing technologies for Internet of things: A primer. Digital Communications and Networks, 2018, 4(2): 77−86.

[11] Aral A, Brandic I. Dependency mining for service resilience at the edge. In: Proc. of the IEEE/ACM Symp. on Edge Computing. ACM, 2018.

[12] Feng Z, George S, Harkes J, *et al*. Edge-based discovery of training data for machine learning. In: Proc. of the IEEE/ACM Symp. on Edge Computing. ACM, 2018.

[13] Porambage P, Okwuibe J, Liyanage M, *et al*. Survey on multi-access edge computing for Internet of Things realization. IEEE Communications Surveys & Tutorials, 2018, 20(4): 2961−2991.

[14] Li JR, Li XY, Gao YL, Gao YQ, Fang BX. Research on data forwarding model in Internet of things environment. Ruan Jian Xue Bao/Journal of Software, 2018, 29(1): 196−224 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5373.htm [doi: 10.13328/j.cnki.jos.005373]

[15] Shi WS, Sun H, Cao J, Zhang Q, Liu W. Edge computing: A new computing model in the era of Internet of everything. Computer Research and Development, 2017, 54(5): 907−924 (in Chinese with English abstract).

[16] Zhang JX, Wu XL, Yang Z, Li J. Research and application of industrial data acquisition technology based on industrial Internet of things. Telecommunications Science, 2018, 34(10): 124−129 (in Chinese with English abstract).

[17] Li SN, Luo GJ. Overview of industrial Internet of things technology and application. Telecommunications Network Technology, 2014(3): 26−31 (in Chinese with English abstract).

[18] Zuo PL, Zhou Q, Dai X. Analysis of industrial Internet of things technology in smart factory. Technology Style, 2019(8): 88.

[19] Zeng J, Li C, Zhang LJ. A face recognition system based on cloud computing and AI edge for IOT. In: Proc. of the Edge Computing (EDGE 2018). Cham: Springer-Verlag, 2018.

[20] Yang YM, Song ZH. Research on security and protection technology of industrial Internet of things. Internet of Things Technology, 2015, 5(3): 64−66 +69 (in Chinese with English abstract).

[21] Sha LT, Xiao F, Chen W, Sun J, Wang RC. Backdoor privacy leakage perception method for industrial Internet of things environment. Ruan Jian Xue Bao/Journal of Software, 2018, 29(7): 1863−1879 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5356.htm [doi: 10.13328/j.cnki.jos.005356]

[22] Liu RL, Liu HT, Xia SF, Yuan D, Wen Y, Cheng LH. Application and thinking of Internet of things technology in power distribution area. High Voltage Technology, 2019, 45(6): 1707−1714 (in Chinese with English abstract).

[23] Cai YM, Feng SY, Du HW, Liu MX, Ding XH, Ji WL. Edge-aware adaptive data processing method for ubiquitous power Internet of things. High Voltage Technology, 2019, 45(6): 1715−1722 (in Chinese with English abstract).

[24] Chen XL, Wan S, Zhu YF, Xu Q. Analysis of distributed power distribution fault processing based on edge computing. Electromechanical Information, 2019(17): 32−33.

[25] Xu Han. Realization of edge computing in motor monitoring system. Electronic Technology and Software Engineering, 2019(11): 190−192.

[26] Saxena H, Salem K. EdgeX: Edge replication for Web applications. In: Proc. of the Int'l Conf. on Cloud Computing. 2015. 1041−1044.

[27] 2019. https://edgent.apache.org/

[28] Zhou Q. Five-year history of GE industrial Internet. China Industry and Information Technology, 2018(7): 32−38 (in Chinese with English abstract).

[29] Jointly Released by Edge Computing Product Alliance (ECC) and Industrial Internet Industry Alliance (AII). Edge Computing Security White Paper. 2019 (in Chinese).

[30] Shen CX. Using active immune trusted computing 3.0 to build a strong network security line of defense and create a clear cyberspace. Information Security Research, 2018, 4(4): 282−302 (in Chinese with English abstract).

[31] Ning ZH. Research on key technologies of trust in the perception layer of the Internet of things [Ph.D. Thesis]. Beijing: Beijing University of Technology, 2016 (in Chinese with English abstract).

[32] Yang WY, Liu W, Huang H, Qi LY, Guo Y. Technical research on power special security operating system based on microkernel. Power Information and Communication Technology, 2016, 14(11): 22−27 (in Chinese with English abstract).

[33] Yadav T, Mallari RA. Technical aspects of cyber kill chain. In: Proc. of the 3rd Int'l Symp. on Security in Computing and Communications (SSCC 2015). 2015.

[34] Yang B, Feng DG, Qin Y, Zhang YJ. TrustZone-based trusted mobile terminal cloud service security access solution. Ruan Jian Xue Bao/Journal of Software, 2016, 27(6): 1366−1383 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5000.htm [doi: 10.13328/j.cnki.jos.005000]

[35] Götzfried J, Eckert M, Schinzel S, *et al*. Cache attacks on Intel SGX. In: Proc. of the European Workshop. 2017.

[36] Villata I, Bidarte U, Kretzschmar U, *et al*. Fast and accurate SEU-tolerance characterization method for Zynq SoCs. In: Proc. of the Int'l Conf. on Field Programmable Logic & Applications. IEEE, 2014.

[37] Qu YW. Software behavior in the network virtual world. Network Security Technology and Application, 2004(4): 15−18 (in Chinese with English abstract).

[38] Veen VVD, Andriesse D, Gkta E, *et al*. Practical context-sensitive CFI. In: Proc. of the 22nd ACM SIGSAC Conf. ACM, 2015.

[39] Fu JM, Tao F, Wang D, Zhang HG. Object-Based software behavior model. Ruan Jian Xue Bao/Journal of Software, 2011, 22(11): 2716−2728 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3923.htm [doi: 10.3724/SP.J.1001.2011.03923]

[40] Jaeger T, Sailer R, Shankar U. PRIMA: Policy-reduced integrity measurement architecture. In: Proc. of the 11th ACM Symp. on Access Control Models and Technologies. New York: ACM, 2006. 10−28.

[41] Gao P. Research on key technologies of RFID middleware based on complex event processing [Ph. D. Thesis]. Beijing: Beijing University of Posts and Telecommunications, 2013 (in Chinese with English abstract).

[42] Lu SB, Zhang M, Lin ZC, Li H, Kuang XH, Zhao G. Function call tracking based on dynamic binary translation and instrumentation. Computer Research and Development, 2019, 56(2): 421−430 (in Chinese with English abstract).

[43] Mak KK. Roadside safety analysis program (Rsap): Engineer's manual. NCHRP Report, 2003.

[44] Liu HF, Qing SH, Liu WQ. Design and implementation of security operating system audit. Computer Research and Development, 2001, 38(10): 1262−1268 (in Chinese with English abstract).

[45] 2020. https://owasp.org/www-project-internet-of-things/

附中文参考文献:

[1] 曹芷晗, 卢煜成, 赖思思, 於志勇, 马樱, 王田. 基于边缘计算的传感云研究进展. 软件学报, 2019, 30(Suppl.(11)): 40−50. http://www.jos.org.cn/1000-9825/19005.htm

[14] 李继蕊, 李小勇, 高雅丽, 高云全, 方滨兴. 物联网环境下数据转发模型研究. 软件学报, 2018, 29(1): 196−224. http://www.jos.org.cn/1000-9825/5373.htm [doi: 10.13328/j.cnki.jos.005373]

[15] 施巍松, 孙辉, 曹杰, 张权, 刘伟. 边缘计算: 万物互联时代新型计算模型. 计算机研究与发展, 2017, 54(5): 907−924.

[16] 张建雄, 吴晓丽, 杨震, 李洁. 基于工业物联网的工业数据采集技术研究与应用. 电信科学, 2018, 34(10): 124−129.

[17] 李士宁, 罗国佳. 工业物联网技术及应用概述. 电信网技术, 2014(3): 26−31.

[20] 杨悦梅, 宋执环. 工业物联网安全及防护技术研究. 物联网技术, 2015, 5(3): 64−66+69.

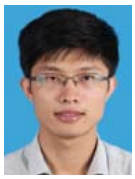
[21] 沙乐天, 肖甫, 陈伟, 孙晶, 王汝传. 面向工业物联网环境下后门隐私泄露感知方法. 软件学报, 2018, 29(7): 1863−1879. http://www.jos.org.cn/1000-9825/5356.htm [doi: 10.13328/j.cnki.jos.005356]

[22] 刘日亮, 刘海涛, 夏圣峰, 袁栋, 文艳, 程力涵. 物联网技术在配电台区中的应用与思考. 高电压技术, 2019, 45(6): 1707−1714.

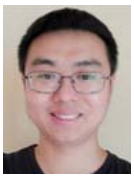
[23] 蔡月明, 封士永, 杜红卫, 刘明祥, 丁孝华, 嵇文路. 面向泛在电力物联网的边缘节点感知自适应数据处理方法. 高电压技术, 2019, 45(6): 1715−1722.
[28] 周倩. GE 工业互联网五年历程. 中国工业和信息化, 2018(7): 32−38.
[29] 边缘计算安全白皮书. 边缘计算产品联盟(ECC)与工业互联网产业联盟(AII)联合发布, 2019.
[30] 沈昌祥. 用主动免疫可信计算 3.0 筑牢网络安全防线营造清朗的网络空间. 信息安全研究, 2018, 4(4): 282−302.
[31] 宁振虎. 物联网感知层可信关键技术研究 [博士学位论文]. 北京: 北京工业大学, 2016.
[32] 杨维永, 刘苇, 黄皓, 祁龙云, 郭毅. 基于微内核的电力专用安全操作系统技术研究. 电力信息与通信技术, 2016, 14(11): 22−27.
[34] 杨波, 冯登国, 秦宇, 张英骏. 基于 TrustZone 的可信移动终端云服务安全接入方案. 软件学报, 2016, 27(6): 1366−1383. http://www.jos.org.cn/1000-9825/5000.htm [doi: 10.13328/j.cnki.jos.005000]
[37] 屈延文. 网络虚拟世界软件行为学. 网络安全技术与应用, 2004(4): 15−18.
[39] 傅建明, 陶芬, 王丹, 张焕国. 基于对象的软件行为模型. 软件学报, 2011, 22(11): 2716−2728. http://www.jos.org.cn/1000-9825/3923.htm [doi: 10.3724/SP.J.1001.2011.03923]
[41] 高鹏. 基于复杂事件处理的 RFID 中间件关键技术研究 [博士学位论文]. 北京: 北京邮电大学, 2013.
[42] 卢帅兵, 张明, 林哲超, 李虎, 况晓辉, 赵刚. 基于动态二进制翻译和插桩的函数调用跟踪. 计算机研究与发展, 2019, 56(2): 421−430.
[44] 刘海峰, 卿斯汉, 刘文清. 安全操作系统审计的设计与实现. 计算机研究与发展, 2001, 38(10): 1262−1268.

杨维永(1978－), 男, 研究员级高工, CCF 专业会员, 主要研究领域为网络安全, 物联网.

黄皓(1957－), 男, 博士, 教授, 博士生导师, 主要研究领域为计算机安全.



刘苇(1986－), 男, 工程师, 主要研究领域为操作系统安全, 云计算安全, 移动安全, 工业信息安全.



廖鹏(1985－), 男, 工程师, 主要研究领域为人工智能, 网络安全.

崔恒志(1971－), 男, 研究员级高工, 主要研究领域为信息化, 电力物联网, 网络安全.

钱柱中(1980－), 男, 博士, 副教授, 博士生导师, CCF 专业会员, 主要研究领域为分布式系统, 数据中心网络.

魏兴慎(1986－), 男, 工程师, 主要研究领域为边缘计算, 物联网, 工业信息安全, 云计算安全.

王元强(1980－), 男, 高级工程师, 主要研究领域为网络安全, 嵌入式系统.