中国科学院
CHINESE ACADEMY OF SCIENCES

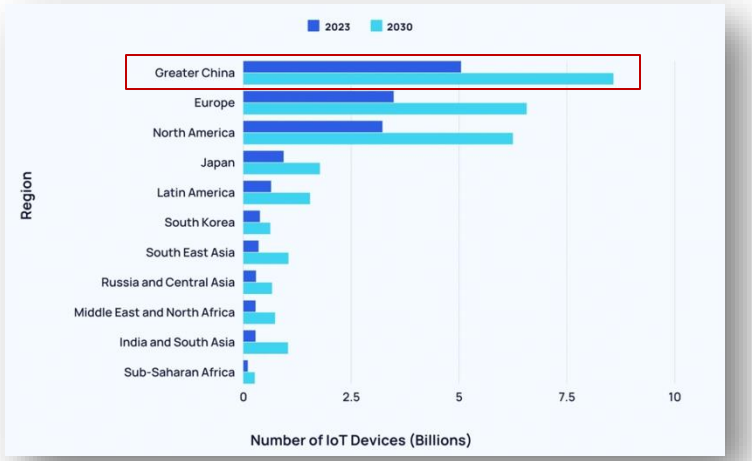
SeChain：基于国密算法的RISC-V安全启动机制设计与实现

芮志清 梅瑶 陈振哲 吴敬征✉ 凌祥 罗天悦 武延军
中国科学院软件研究所 智能软件研究中心
ISCAS 中国科学院软件研究所 Institute of Software Chinese Academy of Sciences
智能软件研究中心 Intelligent Software Research Center

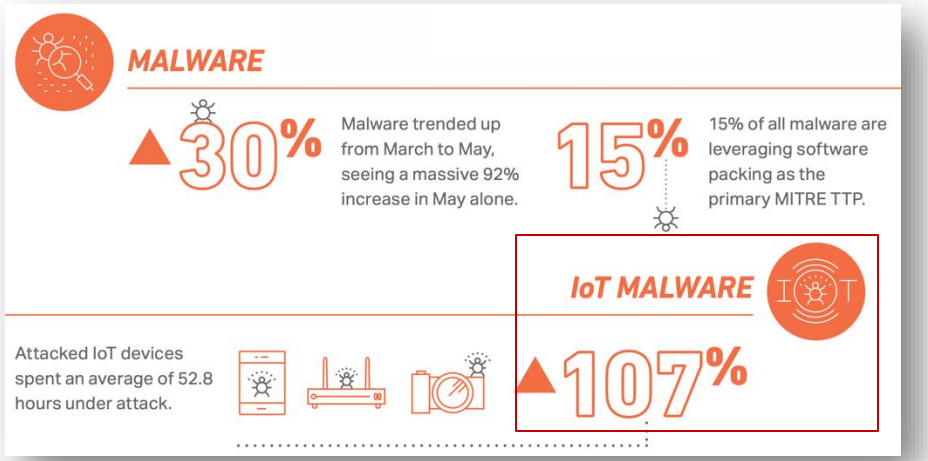
IoT设备数量迅速增加，安全态势严峻

- 据Statista统计，全球IoT设备已于2023年达到159亿，将于2030年实现**翻倍**，达到32.1亿
- 其中，中国的IoT设备数量在全球各区域中贡献**最大比例**，占比约31.7%
- SonicWall报告显示，2024年上半年的**受攻击**的IoT设备数量相比去年同期增长了**107%**

IoT设备连接数量统计及预测
(2022-2033)



全球各区域物联网设备数量
(2022-2033)



物联网攻击数量报告
(2024年上半年)

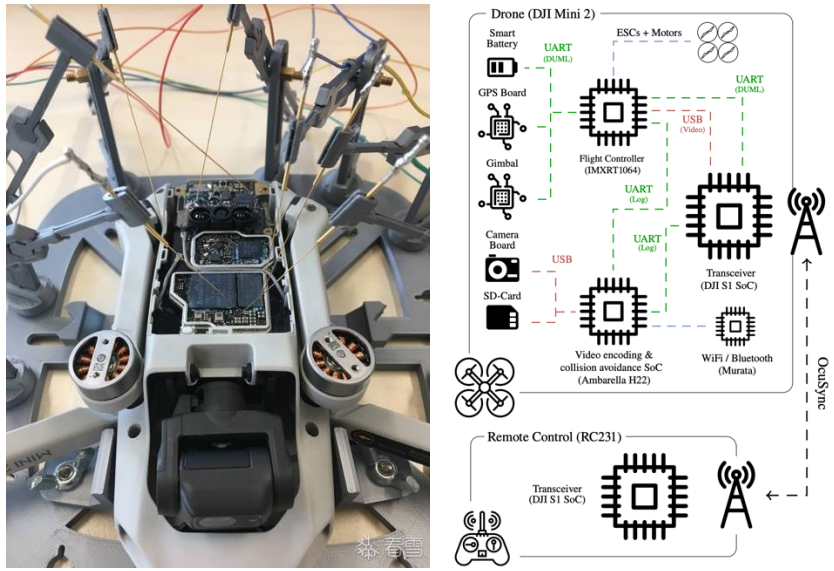
[1] https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/
[2] https://www.sonicwall.com/medialibrary/en/white-paper/mid-year-2024-cyber-threat-report.pdf

缺乏物理层安全机制是IoT设备的重要安全挑战

- 知名安全机构OWASP评选的IoT十大威胁挑战将**缺乏物理层安全机制**作为重要安全挑战
- 存在被物理攻击的隐患，**固件篡改攻击**是其中的典型攻击类型
- 固件篡改攻击：篡改设备原有固件，植入未被授权的代码，实现恶意控制、数据窃取
  - 案例：攻击者基于物理攻击上传固件热补丁，实现对大疆无人机的攻击劫持 [NDSS'23]

OWASP IoT Top10 2018版	OWASP IoT Top10 2024版
Weak Guessable, or Hardcoded Passwords	Insecure Web Interface
Insecure Network Services	Insufficient Authentication/Authorization
Insecure Ecosystem Interfaces	Insecure Network Services
Lack of Secure Update Mechanism	Lack of Transport Encryption
Use of Insecure or Outdated Components	Privacy Concerns
Insufficient Privacy Protection	Insecure Cloud Interface
Insecure Data Transfer and Storage	Insecure Mobile Interface
Lack of Device Management	Insufficient Security Configurability
Insecure Default Settings	Insecure Software/Firmware
Lack of Physical Hardening	**Poor Physical Security**

OWASP IoT Top10

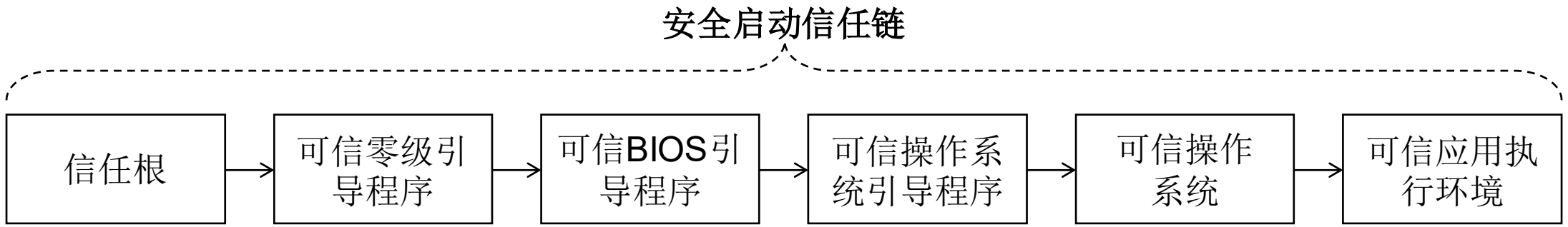


对大疆无人机实施固件篡改攻击

[1] https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10
[2] "Drone Security and the Mysterious Case of DJI's DroneID." NDSS 2023. https://doi.org/10.14722/ndss.2023.24217.

安全启动是解决固件篡改攻击的有效手段

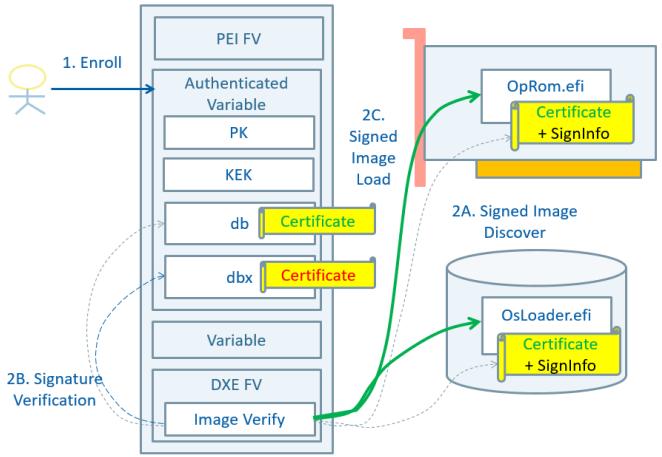
- **安全启动**通过确保设备启动过程中只运行经过授权和验证的软件来保证固件的完整性
- 从信任根出发，逐级顺序验证并加载下一启动阶段的相关程序，以对抗固件篡改攻击



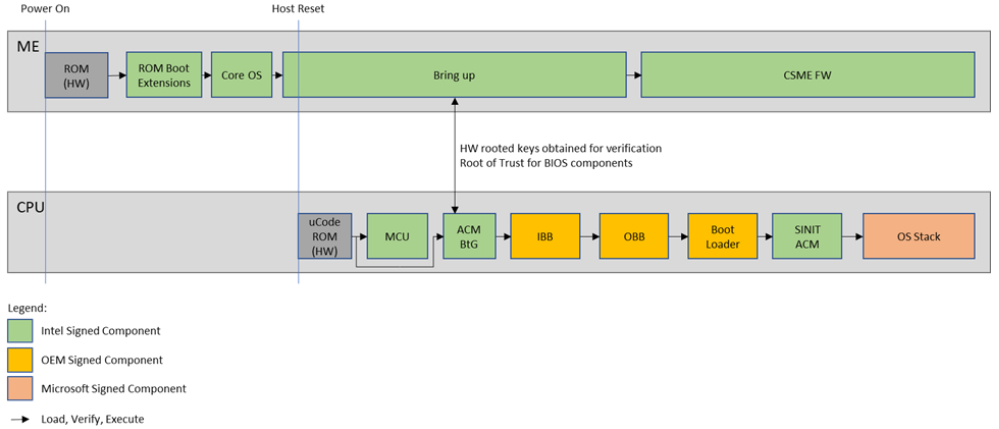
安全启动的信任链传递过程

现有主流安全启动机制

- **UEFI Secure Boot**把OEM固件作为信任根，用于校验第三方固件及程序的完整性
  - 无法保证OEM固件自身的完整性
- **Intel Boot Guard**将CPU中的MCU模块和公钥哈希作为信任根，实现外部固件校验
  - MCU的实现逻辑黑盒，安全审计困难
  - Intel CPU较为重量级，不适用于IoT的轻量级场景
  - 仅Intel和被授权的供应商拥有私钥，无法应用于国内高安全场景



UEFI Secure Boot

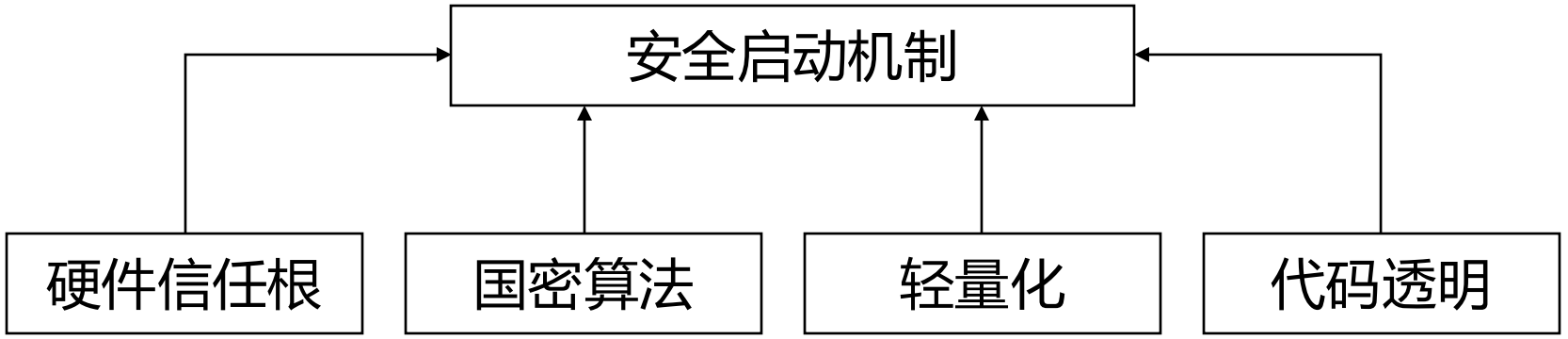


Intel Boot Guard

[1] https://tianocore-docs.github.io/Understanding_UEFI_Secure_Boot_Chain/draft/secure_boot_chain_in_uefi/

研究目标：一种自主可控的安全启动机制

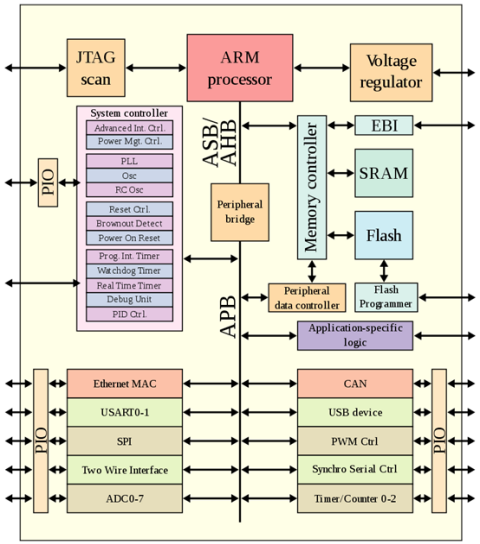
- 设计并实现一种公开透明、自主可控的安全启动机制
  - 基于**硬件**构建**信任根**及校验程序，确保其不可篡改
  - 基于**国密算法**实现密钥计算，可应用于国产高安全场景
  - 实现方式**轻量级**，适用于轻量级的IoT场景
  - 实现原理和对应的代码/电路设计**公开透明**，可被终端用户安全审计



设计思路

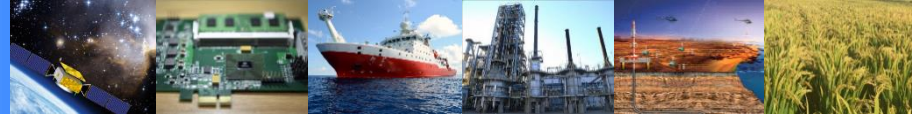
- **开源**、低成本、**轻量级**的RISC-V架构处理器与**公开透明**、**轻量级**的研究目标高度适配
- 将信任根置于片上系统(SoC)中的只读存储单元(ROM)中，基于硬件机制实现**不可篡改**
- 使用**国密SM9**标识密码算法实现非对称加解密

RISC-V核



典型SoC架构

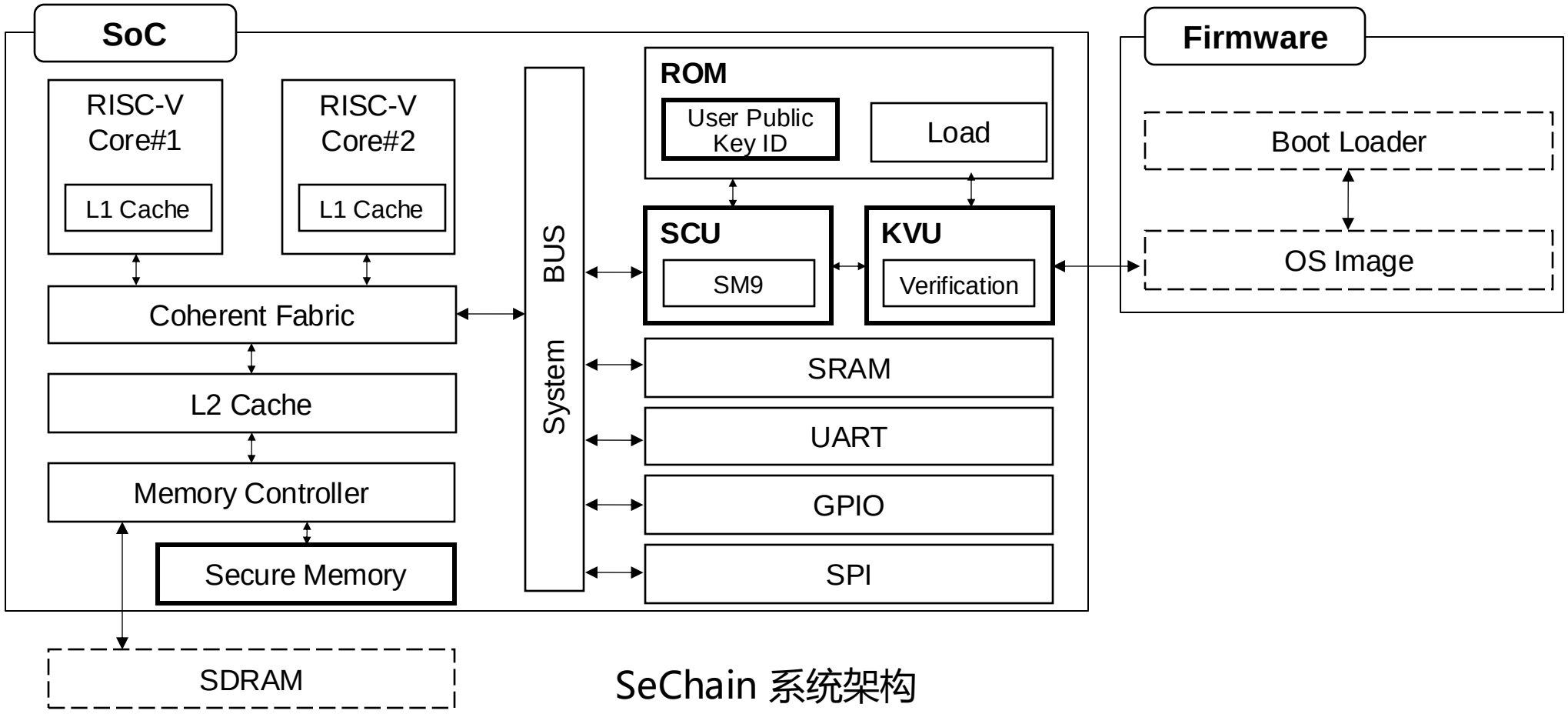
SM9算法标准



架构设计

SeChain系统架构

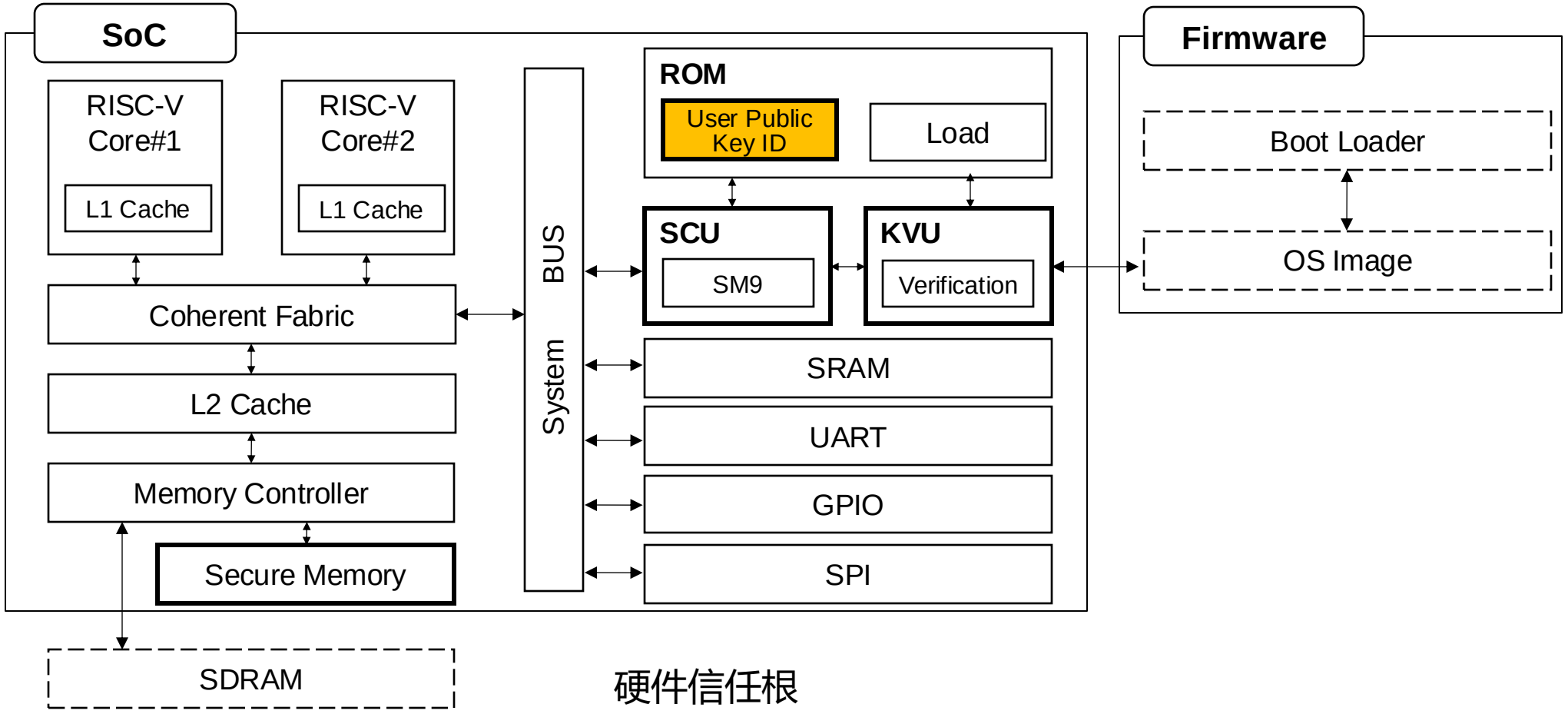
- 在通用RISC-V SoC的基础上，增加签名计算单元(SCU)、密钥验证单元(KVU)，设计实现基于SM9的多级安全启动机制，从不可篡改的硬件信任根出发，逐级完成固件完整性校验



SeChain 系统架构

硬件信任根

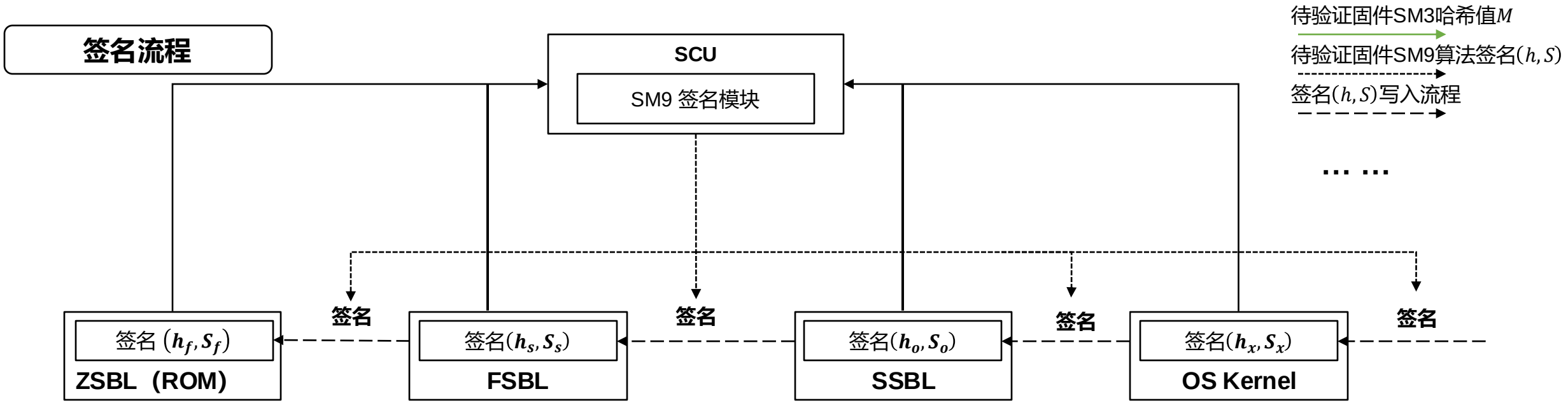
- 设备标识ID作为公钥，烧录在SeChain SoC中的ROM中，只可一次性写入，无法被篡改



硬件信任根

签名流程和SCU单元

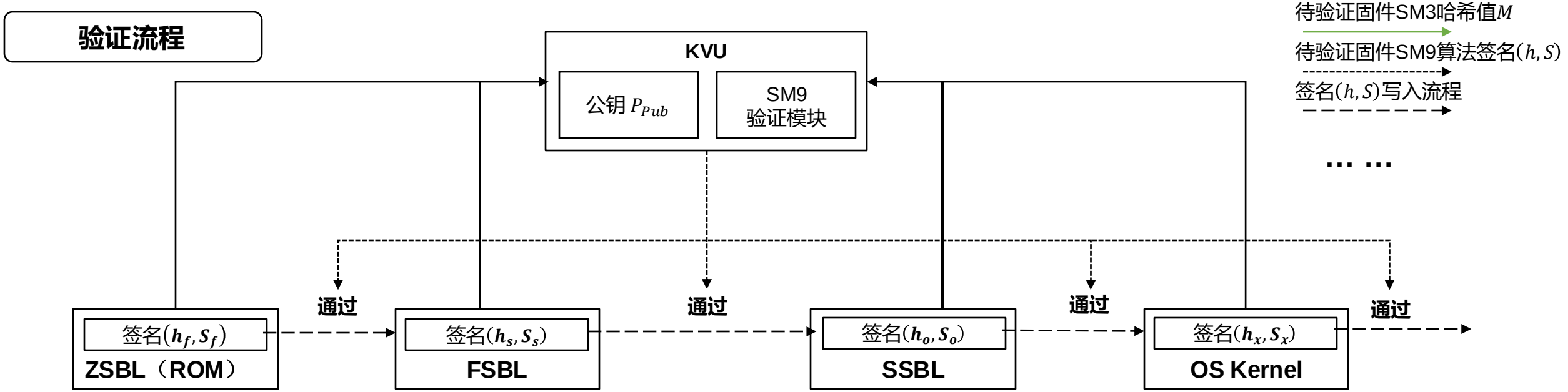
- 增加签名计算单元（signature calculation unit, SCU）
- 设备激活时，SCU根据用户标识ID调用SM9算法生成主密钥对和用户密钥对。随后，SCU使用用户私钥对各级引导程序进行签名，将签名存入上一级引导程序帧头



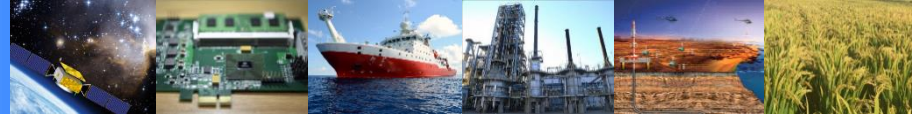
SeChain签名流程

验证流程和KVU单元

- 增加签名计算单元（signature calculation unit, SCU）
- 实现SM9验证算法的片内执行，通过嵌入的用户公钥与签名对固件进行完整性验证



SeChain验证流程

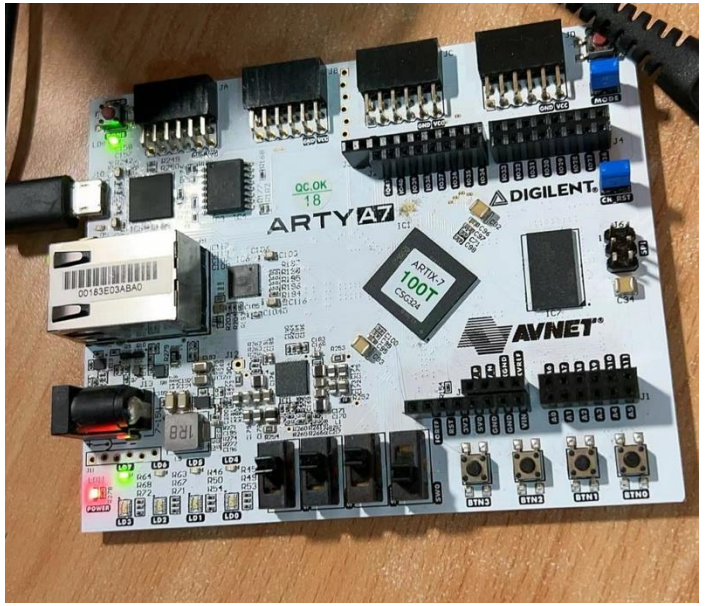


实验验证

实验环境

- SeChain SoC实现：
  - 基础平台：VexRiscv 32bit RISC-V CPU和Litex SoC平台
  - 开发SCU、KVU 模块，增加Secure Memory，修改板载ROM模块和总线配置

- 硬件环境：Xilinx Artix-7 100T FPGA开发版
- 软件负载：OpenSBI 0.8、 Linux Kernel 5.0.9



硬件环境

有效性验证实验

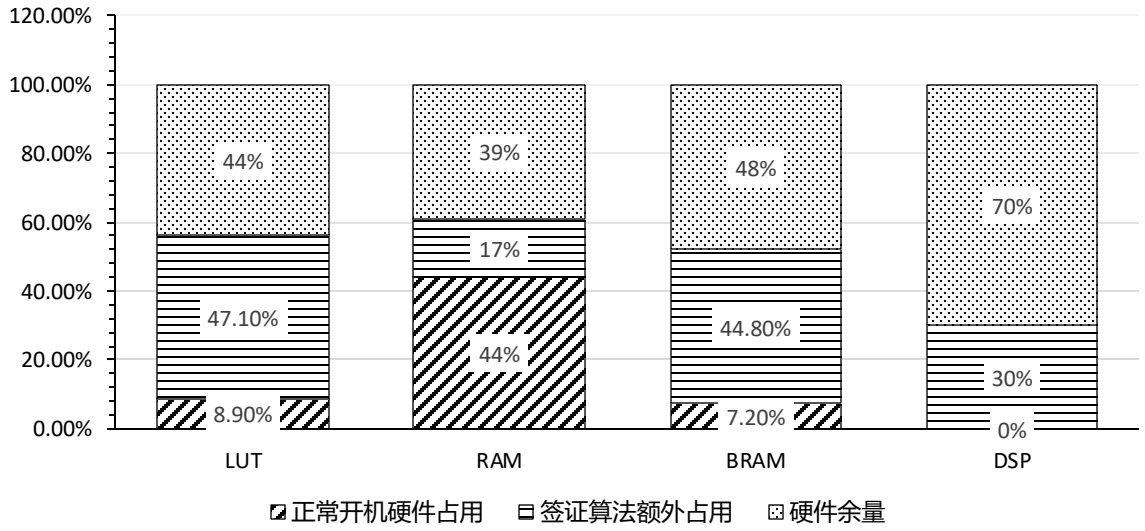
- 实施固件篡改攻击，观察SeChain能否阻断异常固件的启动过程
  - 攻击方式包括：修改跳转地址、修改关键函数、增加额外模块
- 实验结果表明，任意固件更改都将引起KVU 单元签名验证失败，阻断启动进程

攻击位置	攻击方式	攻击检测结果	攻击阻断时间/s	无攻击时验证成功时间/s
boot.c	修改调用 boot helper 函数时传入的地址	验证失败，无法启动	3.1	3.9
	将 boot helper 传入地址硬编码为 0x00000	验证失败，无法启动	3.2	3.9
	修改 boot helper 中所有地址为固定值	验证失败，无法启动	3.2	4.0
	修改 seiral boot 启动时的 printf 函数	验证失败，无法启动	4.1	3.8
	在任意位置插入新的 printf 调用	验证失败，无法启动	4.4	3.9
	删除 serial boot 的 printf 函数	验证失败，无法启动	2.9	3.7
main.c	修改头部关于 litex logo 的 printf 语句	验证失败，无法启动	3.0	3.8
	大面积删除不影响编译完整性的 printf	验证失败，无法启动	3.3	3.9
	修改调用 readline 函数时传入的参数	验证失败，无法启动	3.3	4.0
	删除对 readline 函数的调用	验证失败，无法启动	3.8	3.8
	删除除 serial boot 外的 boot 方式	验证失败，无法启动	3.6	3.9
	删除将是否进入 console 的判断语句	验证失败，无法启动	3.0	3.7
	删除 boot_sequence 函数的调用	验证失败，无法启动	3.8	3.8
readline.c	修改输入对照哈希表	验证失败，无法启动	4.3	3.9
	修改 readline 中对 command 的处理方式	验证失败，无法启动	3.5	3.9
	增加一个新的 command 处理结果	验证失败，无法启动	3.9	3.8
cmd_bios.c	任意位置增加一行 for 空循环语句	验证失败，无法启动	4.2	3.9
	添加 printf 函数调用, 打印某不被支持的 cmd	验证失败，无法启动	3.3	3.7
	删除所有 printf 函数使无法展示可用 cmd	验证失败，无法启动	2.9	3.8
	删除获得 help cmd 后的 printf 函数	验证失败，无法启动	3.6	3.9

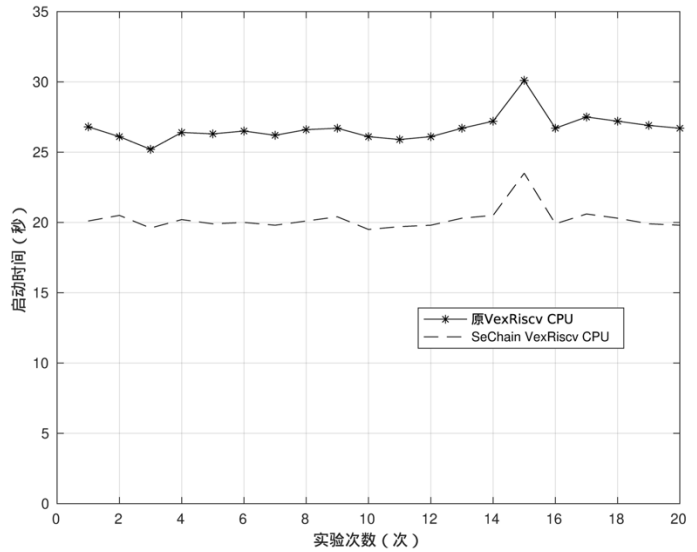
有效性验证实验结果

高效性验证实验

- 对原VexRiscv SoC与SeChain SoC进行了多次不同的启动测试，记录硬件占用和时间开销
- 启动过程中LUT、RAM、BRAM、DSP使用率分别增长了47.10%、10%、44.8%、30%
- 相比原VexRISC-V CPU，SeChain启动时间增长差值仅为6.47s



硬件占用开销图



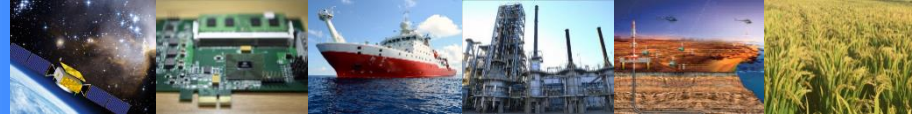
启动时间增长对比图

信任根攻击实验

- 对使用软件和硬件信任根的方案实施攻击，观察设备启动情况，验证硬件信任根的优势
- 实验结果表明，相对于无安全启动和软件信任根的方案，基于硬件信任根的安全启动机制能够**抵御攻击**，阻止被篡改后的固件启动，具有良好的防御效果

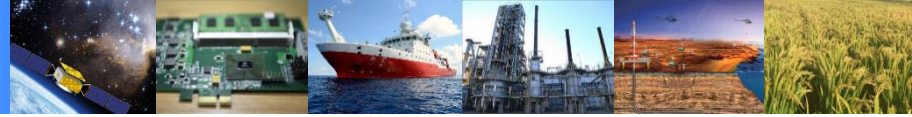
攻击实验	信任链	信任根位置	攻击方式	攻击检测结果	平均时间/s
4a			篡改 boot.c 中的 printf 函数	无验证，攻击成功	3.3
4b	软件	闪存	攻击者窃取设备 *ID*，篡改 boot.c 中的 printf 函数，使用攻击者私钥伪造签名，并修改闪存中的公钥为对应的攻击者公钥	验证通过，设备启动异常，攻击成功	7.3
4c	硬件	写入 ROM	攻击者窃取设备 *ID*，篡改 boot.c 中的 printf 函数，使用攻击者私钥伪造签名，并修改闪存中的公钥为对应的攻击者公钥	验证失败，停止启动，攻击失败	4.3
4d	硬件	写入 ROM	攻击者预先知晓用户私钥与系统参数，篡改 boot.c 中的 printf 函数后，使用用户私钥伪造签名	验证通过，设备启动异常，攻击成功	6.8

信任根攻击实验结果



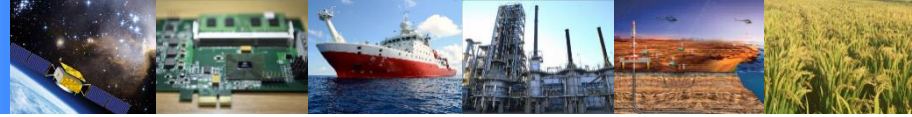
讨论

- 实现了自主可控的安全启动机制，达成硬件信任根、国密算法、轻量化、代码透明的目标
- 相较于传统安全启动机制的**额外优势**：
  - **避免私钥泄漏**
    - 传统机制集中管理私钥，其私钥泄漏会导致海量设备受到固件篡改攻击风险。如2023年5月的MSI的Boot Guard私钥泄漏事件中，攻击者可通过泄漏的私钥签名固件，欺骗原有安全校验机制。
    - SeChain支持终端用户执行初始化过程，生成密钥对和签名，并在签名完成后**将私钥废弃**，避免私钥泄漏导致的安全风险。
  - **无需证书管理**
    - 传统安全启动机制依赖于PKI体系，数字证书由CA签发，用来验证公钥的合法性和绑定关系
    - SM9采用基于标识的密码学，用户公钥可直接从身份信息生成，无需证书
    - 相较于PKI体系的对于完整证书信任链的依赖，SM9算法仅需信任私钥生成中心
    - 信任链更加简化，有利于在IoT场景的轻量化应用



未来工作展望

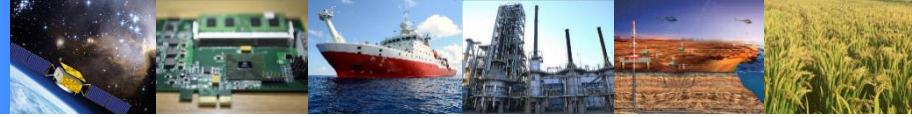
- 增加固件更新机制
  - 当前方法的设备ROM仅可在激活时一次性写入签名，且私钥随即废弃，这在保证高安全性的同时也损失了一定的灵活性。后续研究将在SeChain的基础上提出更优化的固件更新机制，实现固件修改后的签名更新。
- 在真实RISC-V芯片基础上测试评估
  - 受限于时间周期和成本因素，当前方法仅在FPGA平台上实现了仿真验证。后续研究将考虑在真实RISC-V芯片的基础上构建SeChain SoC，并在IoT实际场景上具体应用。



总结

- 提出首个基于SM9国密算法的RISC-V安全启动机制SeChain，将可信根嵌入SoC内部以实现不可篡改性，并设计了SCU和KVU模块，实现了基于验证引导的多级安全启动流程。
- 基于VexRiscv CPU 搭建RISC-V仿真平台，并在FPGA硬件平台进行仿真，在应用SeChain的RISC-V CPU上高效可靠地实现了安全启动。
- 对包含安全启动机制的SeChain原型系统进行有效性、高效性和信任根攻击实验，实验结果表明，该系统能够有效抵御各类固件篡改攻击，验证了其安全启动流程的有效性、高效性和可靠性 。

- **论文**：计算机研究与发展, 2024, 61(6): 1458-1475. DOI: 10.7544/issn1000-1239.202440088
- **代码**：https://github.com/m2kar/SeChain



谢谢!

芮志清
zhiqing@iscas.ac.cn
中国科学院软件研究所