

哈尔滨工业大学深圳校区

毕业论文开题报告

题 目 基于 RISC-V 架构的轻
量级嵌入式安全保护系
统

姓 名 常伯符

学 号 220320221

学 院 机器人与先进制造

专 业 自动化

指 导 教 师 熊昊

日 期 2025.10.13

目 录

1 课题背景及研究的目的与意义	1
1.1 课题背景.....	1
1.2 研究的目的和意义.....	2
1.2.1 研究动机	2
1.2.2 研究意义	2
2 研究现状及分析	3
2.1 国内外研究现状.....	3
2.1.1 RISC-V 安全特性	3
2.1.2 密码学算法与硬件加密	3
2.1.3 侧信道攻击与防护技术	4
2.1.4 后量子密码学	5
2.2 国内外文献综述及简析.....	6
2.2.1 RISC-V 平台安全扩展与验证	6
2.2.2 密码学算法加速与硬件加密设计	7
2.2.3 侧信道攻击与防护技术	8
2.2.4 后量子密码学的硬件实现与安全设计	9
3 主要研究内容及研究方案	10
3.1 项目开发板方案选择.....	10
3.2 研究内容与研究方案.....	12
3.2.1 对称加密算法研究与技术对比	12
3.2.2 后量子签名技术应用与固件完整性检验	13
3.2.3 侧信道防护优化	15
4 进度安排及预期目标	16
4.1 进度安排.....	16
4.2 预期目标.....	18
5 已具备和所需的条件和经费	21
5.1 实验室条件和经费保障.....	21
5.2 所需条件和经费.....	21
6 预计困难及解决方案	22
6.1 CM32M433R 官方文档部分缺失、信息过时	22

6.2 SAC 模块配置复杂，可能会初始化失败.....	22
6.3 硬件资源受限.....	22
6.4 指标测定方法过于理想化，实验室条件不足.....	23
6.5 SLH-DSA 性能低，需求计算资源较多	23
6.6 系统集成的潜在困难.....	23
7 参考文献	24

1 课题背景及研究的目的与意义

1.1 课题背景

物联网（Internet of Things, IoT）的蓬勃发展正改变着世界，一个崭新的物联网时代已然来临。据统计[1]，截至 2025 年，超过 198 亿台 IoT 设备已连入物联网，且预估截至 2034 年将会有超过 406 亿台设备，其中中国将拥有超过 75 亿台消费级设备，成为 IoT 设备数目最多的大国。然而，如此多的轻量级设备暴露在复杂的物理环境、社会环境和网络环境中，势必会带来诸多安全风险。而且，IoT 设备由于成本、功耗、体积等多方面因素的限制，往往无法直接复用个人电脑或服务上的成熟安全方案。因此，轻量级 IoT 设备往往成为系统安全的薄弱环节。

精简指令集 RISC-V（Reduced Instruction Set Computer V）凭借其开源性、模块化、轻量级、免版税等特性，成为 IoT 设备的主流架构选择。在安全方面，RISC-V 支持 Machine / Supervisor / User 三种特权模式（Privileged Mode）、物理内存保护（Physical Memory Protection, PMP）、Crypto 扩展等，可按需添加、裁剪模块，充分适配轻量级场景，为 IoT 设备的安全提供保障。这些安全特性将在 2 作进一步讨论。

然而，即使已经拥有了大量的安全特性，很多 IoT 设备和 RISC-V 轻量级设备仍然面临着诸多攻击方式，无论是来自配置上的疏忽还是硬件上的漏洞。例如，在 RISC-V CVA6 项目中[2]，MMU 配置包错误设置 64 个 PMP 条目，而硬件仅支持 16 个，导致权限隔离失效，这是很典型的由于 PMP 配置错误导致应有的安全特性反而成为漏洞的源头的例子。在固件篡改方面，2023 年 HP 公司发现旗下的 T430 和 T638 瘦客户机的私钥被泄露，导致相关固件遭到篡改，该漏洞已被收录至 CVE-2023-5409[3]。这些 BIOS 镜像私钥本被用来提供固件完整性和系统正常使用的额外保障，但一旦遭到泄露，便成了攻击者篡改固件的可乘之机。自从 2018 年著名的 Meltdown 和 Spectre 漏洞被谷歌团队公布以来，侧信道攻击（Side-channel Attack, SCA）等基于硬件层的攻击愈发频繁多样；2024 年，我国 CNCERT 国家工程研究中心联合西北工业大学研究团队首次挖掘出 RISC-V SonicBOOM 处理器上的寄存器端口争用漏洞[4]，验证了 RISC-V 平台同样存在微架构级攻击风险。

虽然 RISC-V 国际基金会和各大厂商都在积极推进 RISC-V 安全性能的进步，但大多数研究集中在高性能、超标量 RISC-V 处理器上，对于中低端微处理器（Micro Control Unit, MCU）的实证研究相对匮乏，缺乏硬件与软件安全方案的定量比较与最佳实践指导。

1.2 研究的目的和意义

1.2.1 研究动机

正如上文所述，大多数 RISC-V 安全研究侧重于高性能应用处理器上（如 OpenTitan），因为这些处理器性能高、受众广，对其进行研究显然更加必要；而对于中低端 MCU 平台的安全评估和防护研究较少，这些 MCU 虽然性能较低、受众较少，但较高性能处理器而言具有一定的成本、功耗、体积优势，而这些方面都是 IoT 设备所重视的指标。因此，若能开展针对中低端 MCU 平台的安全评估和防护研究，则可促进国内外中低端 IoT 设备的开发，进而促进市场良性循环，技术进一步迭代提升。

工业界常面临安全与性能的平衡问题：纯软件实现的加密（如 tiny-AES-c 库）固然便于移植，但性能较低、占用存储空间较大，且存在较多的安全隐患，侧信道安全没有保障；而硬件实现的加密虽然具有更高的性能和更低的存储空间，但刚需配置相应的设备（如密码算法硬件加速引擎 SAC、TRNG 熵源），可移植性难以保证，且方案相对复杂，缺乏最佳实践指导。

另外，传统软件方案难以防御侧信道等硬件层威胁，迫切需要结合 RISC-V 原生特性构建高效、低功耗的防护体系，如特权隔离等安全特性。

1.2.2 研究意义

通过研究，本项目可提供适用于中低端 MCU 的安全实证数据。具体地，通过在 CM32M433R 平台上实现硬件方案（SAC 算法加速，PMP 保护敏感寄存器）和软件方案（通过各种开源库实现可移植的 AES、Ascon、SLH-DSA 算法）并量化其安全指标与性能指标，本项目可为轻量级 RISC-V 设备的安全方案提供可复用的框架，填补中低端 RISC-V MCU 安全防护研究的空白。

通过对比数据，本项目可提供不同场景下方案选择的建议，提供可复用的安全模块代码（如硬件 AES 配置、PMP 权限隔离配置、Bootloader 验证流程）、量化对比数据 and 实践指南，促进自主可控物联网安全生态发展。

最重要的是，通过大四一年的实践，本项目可深化我对 RISC-V 架构的理解，实践 M/U 特权模式、PMP 配置等原生安全特性，掌握轻量级安全的底层逻辑；通过软件加密理解 AES、Ascon 等算法的原理，通过硬件加密掌握外设寄存器配置，最终形成理论-实践闭环，提升嵌入式安全开发能力，为未来更广、更深层次的研究打好基础。

2 研究现状及分析

2.1 国内外研究现状

2.1.1 RISC-V 安全特性

大多数 RISC-V 安全特性设计都基于 RISC-V 特权体系结构(RISC-V privileged architecture) [28], 这套权级设计基于 RISC-V 的模块化设计, 可以完全替换成其他权级设计而无需改变基础 ISA, 从而保留 RISC-V 的高自由度。文档在 v1.10 引入一个可选的物理内存保护 (Physical Memory Protection, PMP) 设计, 使得用户能直接控制指定寄存器的读/写/执行权限, 在特权隔离的基础上提供更加定制化的保护机制。这些特性共同构成 RISC-V 的基础安全隔离机制, 大多数 RISC-V 安全研究都围绕这些机制, 如 PMP 的形式化验证与硬件实现[29]、漏洞分析与防护措施[30]等, 使得 RISC-V 安全特性在实践中得到认证。

RISC-V 基金会也在大力推进 RISC-V 安全特性的升级和增强, 如在 2025 年的 RISC-V 中国峰会上[31], 提到了 RVA23 Profile 的诸多新特性, 其中 IOMMU 与 PMP 的三级嵌套标志着 RISC-V 安全得到有力保障。

在国内, 由于 RISC-V 开源、免版税等特性及受到来自美国的科技制裁等影响, RISC-V 一直是国内相关研究人员和厂商的热门研究领域, 安全特性也不例外。闫润等人[5]探索了特权架构配置在不同应用场景下对功能和硬件资源开销的影响; 王杰等人[6]基于 PMP、Flash 锁定的静态数据保护等机制实现了 RfTPM, 一种基于 RISC-V 的固件可信平台模块架构。随着国内各大研究机构和硬件厂商加入到 RISC-V 的研究中来, RISC-V 安全在可预见的未来将得到持续发展。

2.1.2 密码学算法与硬件加密

2001 年末, 美国国家标准与技术研究院 (National Institute of Standard and Technology, NIST) 发布了著名的高级加密标准 (Advanced Encryption Standard, AES), 又称 Rijndael 加密法, 作为旧标准 DES 的替代。

Marshall 等人[28]研究了如何高效地实现 AES-GCM, 他们的工作最终成为了 RISC-V 密码学指令扩展 (RISC-V Cryptography Extension, K 扩展) 标准化进程的重要内容。K 扩展涵盖标量和熵源指令的密码扩展方案, 包括比特操作、标量 AES、SHA、SM3 和 SM4 加速以及 TRNG 熵源接口; 后基于同样标准化的向量指令扩展 (V 扩展), 实现了向量密码学指令集, 适用于在较大的核心中高效运行。标量 K

扩展于 2021 年 10 月完成公众评论，于 2023 年 10 月正式获得基金会批准成为 RISC-V 标准。不无遗憾的是，K 扩展起步、标准化较晚，目前市面上鲜有支持 K 扩展的低成本 RISC-V 开发平台，故本项目采用集成的异构硬件加密加速模块来代替 K 扩展的功能。

自 2016 年起，NIST 举办了 NIST 轻量级加密算法大赛，旨在寻找适用于低功耗设备的高安全性、高效、抗侧信道攻击的轻量级加密算法和哈希算法。最终，由 Dobraunig 等人设计的 Ascon 算法家族[32]脱颖而出。2023 年，一份名为 NIST SP 800-232[33]的报告正式将 Ascon 算法家族标准化，成为 NIST 针对资源受限的设备的高性能密码学解决方案。

我国密码学算法和硬件加密也有长足发展。上文提到的 SM3、SM4 实际上就是由我国国家密码局公开并大力推广的国产商密（Shangmi）算法，目前已公开的有 ZUC、SM2、SM3、SM4、SM9。在商密产品认证过程中，原则上都要使用商密算法，尽量避免使用国际算法。相关的研究与应用也很丰富，例如秦体红等人[7]基于 SM9 算法设计了一种具有同态性质的加密方案，具备选择明文攻击安全性。另外，针对 RISC-V 上的 AES 指令扩展也有研究，例如杨万[8]通过系统性地研究 RISC-V 处理器核和 AES 算法流程设计了一个 AES 加密单元，实现更佳的性能和更少的指令周期数。此外，国内成立了中国国家信息安全脆弱性数据库（CNNVD）、国家计算机网络与应急技术处理协调中心（CNCERT）等多家安全机构与研究中心，这里不再赘述。

2.1.3 侧信道攻击与防护技术

自 2018 年谷歌团队公布了著名的“熔断”（Meltdown, CVE-2017-5754）和“幽灵”（Spectre, CVE-2017-5753/CVE-2017-5715）以来，隐蔽通道信息泄露和侧信道攻击（缓存、分支预测、时间差异、瞬态执行漏洞等）使得社会各界高度重视硬件级安全[34]。一般来说，侧信道攻击的特点是实现难度较高、需要攻击者物理接触硬件、高精度测量等，然而随着越来越多样化的侧信道攻击方式被发现，这些特点也不再准确，例如 2021 年发现的基于浏览器的侧信道攻击使得攻击者不需要物理接触硬件[9]。

RISC-V 系统自然也难免此类攻击。Fadiheh 等人[10]展示了隐蔽信道泄露是广泛存在的，不仅在先进的基于分支预测的超标量处理器上，也存在于一般复杂度的顺序执行处理器上。他们随后设计了一种可自动检查并定位侧信道薄弱处的方案，并成功应用在 RISC-V Rocket 处理器上。RISC-V 社区与研究者先后提出并实现多种防护模型与措施，例如硬件/软件掩码、洗牌（shuffling）、控制流完整性等，

用以减轻侧信道威胁。其中，掩码技术因其理论安全性保障而被广泛采用，许多硬件设计集成一阶或高阶掩码来抵抗功耗分析攻击。针对缓存、分支预测、互连总线等微架构通道的防护，也越来越受到重视。

由于侧信道攻击主要源于硬件设计漏洞，Xu 等人[11]在 Kyber 硬件设计中引入紧凑的 shuffling 架构，以在保持性能的同时增强抗泄露性。其他研究包括动态部分重配置、实时重构、功能切换等，避免呈现物理内存有规律的模式，从而干扰攻击者构建稳定的模型。

总体来看，侧信道攻击防护研究已经从早期的功耗差分分析、电磁场能量差分分析扩展到复杂微架构通道与高级攻击模型，甚至引入了机器学习来快速辨认时间侧信道信息[12]；而防护策略也从静态、单一方向到动态、混合方向演进，包括防护开销、高阶防护设计、验证方法与标准化评估等都在如火如荼地展开研究。我国在这一领域虽然研究较少，但也有一定的成果，例如前文提及的 CNCERT 联合西北工业大学研究团队首次挖掘出 RISC-V SonicBOOM 处理器上的寄存器端口争用漏洞[4]便属于侧信道攻击的范畴。

2.1.4 后量子密码学

传统加密算法，尤其是公钥密码学（Public-key Cryptography, PKC）的可信度大多源于三个计算难题：整数分解问题、离散对数问题、椭圆曲线离散对数问题。然而，这些难题均可使用量子计算机并应用秀尔算法在多项式时间内破解，这是由数学家彼得·秀尔在 1994 年发现的[13]。虽然直至 2023 年，量子电脑的性能仍无法破解一般使用的加密算法，但密码学家已经在考虑所谓的 Q-Day，即量子电脑能破解传统加密算法的时刻。从 2006 年起举办的一系列 PQCrypto 等研讨会开始，后量子密码学（Post-quantum Cryptography, PQC）的研究已受到学术界和工业界的广泛注意。

后量子密码学的研究方向包括格密码学、容错学习问题、多变量密码学、散列密码学、编码密码学、超奇异椭圆曲线同源密码学等。其中，散列密码学具有安全性仅依赖于哈希函数的优点，而后者已经过长期实践验证，被认为具有高度稳固的安全性基础。许多后量子散列签名方案为有状态类型，可能导致私钥泄露，而在 2015 年提出的 SPHINCS+通过更大的签名体积、更高的计算开销成功实现了无状态散列方案。同年，NIST 启动了抗量子算法的筛选和标准化工作，SPHINCS+算法入选，并在改进后更名为 SLH-DSA，即无状态散列数字签名算法的简称。2024 年 8 月，NIST 发布 FIPS 205[35]，使得 SLH-DSA 算法正式成为 NIST 后量子密码标准之一。

SLH-DSA 的安全性仅依赖于所使用的哈希函数，其算法核心是重复哈希，纯软件实现虽然耗时，但可通过硬件加速模块优化性能表现。Saarinen 等人[36]通过在一个小的 RISC-V 核心的基础上引入一个哈希计算模块并应用 SLH-DSA 算法，实测得到高达两个数量级的性能提升。国内虽暂无将 SLH-DSA 算法应用到 RISC-V 核心上的实例，但也有针对 SLH-DSA 的应用和优化，例如吴晓杰[14]通过输入预处理、展开迭代过程等方法加速算法流程，在 FPGA 上实现 8 个哈希函数核心并行计算，最终对于 SHA-2 哈希函数软件实现 7.5 倍的加速比。

从时间上看，国内外针对 RISC-V 安全领域的研究丰富多彩，RISC-V 安全体系正从标准与框架建设阶段逐渐进入实证与优化阶段；然而，在中低端 MCU 这类资源大幅受限的场景中实现 RISC-V 安全特性的研究稍有不足，面对新型攻击如侧信道攻击与量子电脑仍显乏力，还有较大的研究空间未被覆盖。

2.2 国内外文献综述及简析

2.2.1 RISC-V 平台安全扩展与验证

在 2021 年，Tao Lu 发表了一篇综述类文章[34]。这篇文章涵盖了 RISC-V 硬件和物理访问安全、硬件辅助安全单元、ISA 安全扩展、内存保护、密码学原语和侧信道攻击防护等领域，并对当前的技术瓶颈与未来趋势做出预测。文章首先回顾了 RISC-V 的架构与发展，然后归纳硬件安全模块（HSM）、物理访问防护、ISA 安全指令扩展和密钥存储机制，最后指出未来应加强形式验证和跨层协同安全设计。文章虽然发布于 2021 年，存在一定滞后性，如文中提到的 RISC-V Cryptography 扩展（K 扩展）已在 2023 年获得 RISC-V 基金会批准[37]，成为 RISC-V 非特权扩展指令集标准的一员；但文章系统地阐述了 2021 年及以前的 RISC-V 安全领域的进展、成果和挑战，为后续研究搭建起基本的学术框架。

由 Tain 等人[38]发表的综述类文章则对各类 RISC-V 处理器上的验证方法（形式验证、模拟验证、混合验证、覆盖驱动测试等）及应用进行了深入总结，探讨如何在具有用户自定义扩展指令集的环境下仍能保持可验证性。文章指出，用户扩展指令集（如自定义的加密扩展）极大地增加了验证复杂性，需要新的验证策略与工具支持。

此外，在 RISC-V 架构安全服务支持（文件加密、密钥管理、远程认证等）方向，Bove[39]指出标准 RISC-V ISA 已具备部分安全服务支持能力，但针对安全文件存储、远程认证等安全服务仍存在指令集扩展需求。Boubakri 等人[40]则通过实现一个适用于 RISC-V 处理器的纯软件可信平台模块（Trusted Platform Module,

TPM),在不用调整处理器的微架构的基础上进一步拓宽 RISC-V 安全能力的边界。

总体来看,当前 RISC-V 安全扩展与验证研究以理论与设计为主,未来需要更多从算法-硬件-系统三方去协同设计与验证工作。

2.2.2 密码学算法加速与硬件加密设计

作为加密/解密的核心,密码学算法的安全性、时间空间复杂度和执行效率非常重要,几乎决定了安全特性的性能指标。

2001 年末,美国国家标准与技术研究院(National Institute of Standard and Technology, NIST)发布了著名的高级加密标准(Advanced Encryption Standard, AES),又称 Rijndael 加密法,作为旧标准 DES 的替代。Lu 在文章中指出[34],得益于其高安全性、高性能和简单的密钥管理, AES 至今仍是最为广泛应用的对称加密算法,安全高效地实现 AES 是大多数计算平台的基本要求。

多数平台都通过扩展指令集(Instruction Set Extension, ISE)实现高效的硬件 AES 加解密执行, RISC-V 虽然起步较晚,但也已经基本完成了相关的开发。Marshall 等人[28]调研了 5 种最新的工业界和学术界的 ISE。他们发现,相较于基于 T-table 的纯软件实现,在 32 位 RISC-V 架构上使用 ISE 实现 AES 可达到 4 倍的 AES 块加密性能提升,而只占用了 1100 余个门电路,在 64 位架构则能达到 10 倍提升,占用 8200 余个门电路。他们同样研究了如何利用已有的 RISC-V 标准比特操作指令扩展(B 扩展)高效地实现 AES-GCM。

Marshall 等人的工作最终成为了 RISC-V 密码学指令扩展(RISC-V Cryptography Extension, K 扩展)标准化进程的重要内容。K 扩展涵盖标量和熵源指令的密码扩展方案,包括比特操作、标量 AES、SHA、SM3 和 SM4 加速以及 TRNG 熵源接口;后基于同样标准化的向量指令扩展(V 扩展),实现了向量密码学指令集,适用于在较大的核心中高效运行。标量 K 扩展于 2021 年 10 月完成公众评论,于 2023 年 10 月正式获得基金会批准成为 RISC-V 标准。不无遗憾的是, K 扩展起步、标准化较晚,目前市面上鲜有支持 K 扩展的低成本 RISC-V 开发平台,故本项目采用集成的异构硬件加密加速模块来代替 K 扩展的功能。

近年来,随着 IoT 设备的大规模增长,由于其轻量化的要求,即使是高效如 AES 的加解密算法也显得吃力,甚至无法正常运行。为此,自 2016 年起, NIST 举办了 NIST 轻量级加密算法大赛,旨在寻找适用于低功耗设备的高安全性、高效、抗侧信道攻击的轻量级加密算法和哈希算法。最终,由 Dobraunig 等人设计的 Ascon 算法家族[32]脱颖而出。无论是在关联数据的认证加密还是哈希基准, Ascon 在加密速度、哈希性能、能耗比等多个指标上均有突出表现。2023 年,一份名为 NIST

SP 800-232[33]的报告正式将 Ascon 算法家族标准化，成为 NIST 针对资源受限的设备的高性能密码学解决方案。

总体来看，传统密码学算法如 AES 虽然具有高安全性，但无法很好地适配资源受限的 IoT 设备；新型算法如 Ascon 则填补了 IoT 设备密码学算法的空白，不过其实践效果仍有待验证。本项目将同时集成 AES 与 Ascon 算法，并基于数据对比二者在中低端 MCU 上的性能差异。

2.2.3 侧信道攻击与防护技术

现代处理器的侧信道漏洞使得硬件安全成为处理器设计的重中之重。Gonzalez 等人[15]展示了开源的 RISC-V 超标量处理器——伯克利乱序机(Berkeley Out-of-Order Machine, BOOM)如何在微体系结构层面缓解侧信道攻击。首先，他们复制了 Spectre 漏洞系列中利用 L1 数据缓存中的推测执行的几个基本变种；然后，他们针对这类攻击实施初步的硬件缓解措施，证明其有效性，并衡量其对性能和面积大小的影响。与对照处理器相比，硬件缓解评估表明在 45nm 工艺下，IPC 增加了约 2%，面积增加了约 2.5%，时钟频率减少了约 0.36%，但成功实现了对上述漏洞的阻拦。该工作展示了开源 RISC-V 硬件生态系统对于安全研究者们研究现代处理器的硬件安全漏洞与缓解措施的价值。

纯软件实现的加解密算法往往易受能耗分析和电磁场辐射分析的侧信道攻击。Mulder 等人[16]提出将他们的侧信道防护方案集成到 RISC-V 实现中。该方案在密钥写入内存之前使用掩码技术对其处理一次，并在将密钥从内存中读回后做一次逆向操作对其解密，从而阻止 SCA 对内存分析导致泄密的可能。该方案在防止一阶能耗或电磁攻击的同时，保持尽可能低的实现成本。评估结果证实了各种加密原语在受保护的硬件平台上运行的安全性。

在设计阶段对软件安全漏洞进行评估，可以在早期阶段就发现问题，避免后期漏洞修复的成本，因此非常关键。Sauvage 等人[41]描述了一个通过标量乘法实现的虚拟原型，旨在保护目标平台免受简单的侧信道攻击。他们使用 Mentor Graphics Modelsim 工具来获得尽可能接近现实的信息泄漏再现。该方案模拟软件在开源 32 位 RISC-V 微机 PULPino 上执行，代价则是要求位与时钟周期的较高精度。对于每个时钟周期，他们计算进入微控制器的位数，功耗图像，并观察程序计数器来识别执行的汇编指令，然后识别相应的 C 函数。虚拟分析指出，一边的 double 函数和另一边的 add 函数在管理变量和内部操作的方式上存在差异，这些差异可被攻击者用来提取密钥。当然，该方法尚不成熟，要得到实际应用还面临诸多挑战，如提高仿真性能、更真实的攻击模型、更自动化的部署等。

总体来看，侧信道攻击防护领域仍是一个生机勃勃的研究方向，不断有新型的侧信道攻击方式诞生，也不断有更好的抗侧信道措施出炉。因此，很难说某一设计能“完全抵御侧信道攻击”，但也不能得出“侧信道攻击无法彻底缓解”的结论。作为开源社区的代表，RISC-V 社区及其特性将不断鼓励更多研究者投入到侧信道攻防的研究中。

2.2.4 后量子密码学的硬件实现与安全设计

在 NIST 推出包括 SLH-DSA 在内的多个标准化后量子加密算法之前，后量子密码学的研究分散在诸多领域中，各种研究往往保留一定的灵活性以适配未来的标准化算法。在此背景下，Fritzmenn 等人[42]提出了 RISQ-V，一种增强的 RISC-V 架构，它集成了一组强大的紧耦合加速器来加速基于格密码学的后量子加密算法。RISQ-V 有效地复用了处理器资源，减少了访存量，因此在保持较低芯片硅面积开销的同时显著提高了性能。首先，他们将一套功能强大的硬件加速器深度集成到 RISC-V 流水线中。他们随后增加了 29 条新指令，以在硬件层次上高效地实现了基于格密码学的算法。最后，他们在 FPGA 上实现了 RISQ-V，并评估了 NewHope, Kyber 和 Saber 的性能。与在 RISC-V 上的纯软件实现相比，RISQ-V 分别呈现出 11.4 倍，9.6 倍和 2.7 倍的速度提升，而能耗分别降低至 1/9.5，1/7.7，1/2.1。

SLH-DSA 成为后量子加密算法标准后，相关研究也逐渐向其靠拢。2024 年，Saarinen 等人[36]证明了从硬件上优化 SLH-DSA 算法中的填充格式和迭代哈希过程可以获得非常显著的整体性能增益。他们设计并实现了 SLoth 硬件加速核，包含 Keccak/SHAKE、SHA2-256 和 SHA2-512 内核，支持 SLH-DSA 的全部 12 个参数集，还支持侧信道安全的 PRF 计算和 Winternitz 链。他们将 SLoth 驱动运行在一个小型的 RISC-V 控制核心上，这种搭配在当前的信任根(Root-of-Trust, RoT)系统中很常见。这些新特性使得 SLoth 上的 SLH-DSA 比同等大小的通用哈希加速器快了许多倍。与未加速的微控制器实现相比，SLoth 的 SHAKE 变体的性能提高了至多 300 倍；128f 参数集的签名生成周期为 4903978 轮，而 128s 参数集的签名验证周期仅为 179603 轮。SLoth 占用芯片面积较小，其范围从 63kGE(SHA-2,仅 Cat1)到 155kGE(所有参数)。作为对比，仅 Keccak 阈值实现就额外增加了 130kGE。

总体来看，PQC 领域尚处于激烈的攻防阶段，但自 NIST 标准发布后，相关研究也渐渐向标准化靠拢。然而，这些研究多在 FPGA 平台上实践，应用到 RISC-V 平台的实例较少，SLH-DSA 在 RISC-V 平台上的硬件加速研究仍有不少空白。不过从标准层面上看，SLH-DSA 已经取代了传统的非对称加密算法（如 RSA），相关 IoT 设备若已支持 RSA 算法，须逐渐过渡到 SLH-DSA。

3 主要研究内容及研究方案

3.1 项目开发板方案选择

表格 1

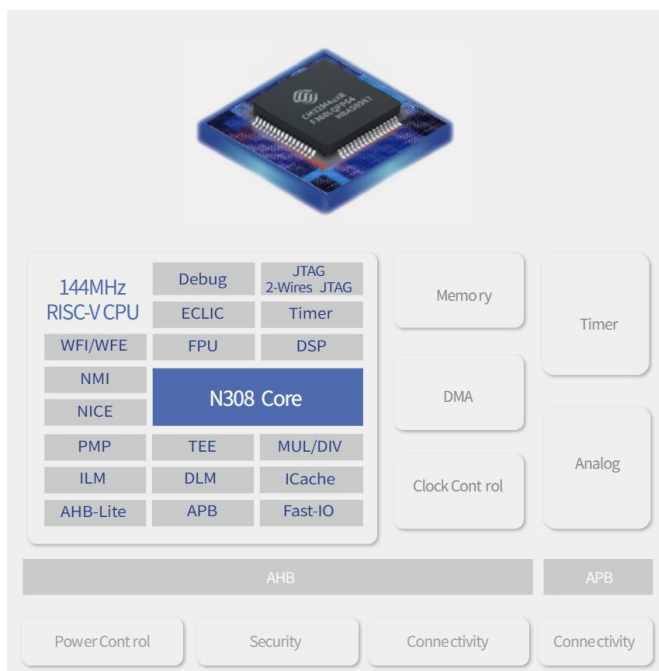
项目	CM32M433R	KD233 (K210)	GD32VF103+ ATECC608B	HiFive1 Rev B	STM32U5 (U585)
参考价格	112 元	180-350 元	27+35 元	停产	约 580 元
典型功耗	取决于频率 与外设	典型功耗 \approx 0.3W, 运行 AES 时 \approx 0.5W[18]	ATECC608B: 睡眠时可 <150nA[17]	与供电能 力和频率 有关	U5 系列属 超低功耗 [19]
AES-128	片上 SAC 支 持 AES 等多 个算法, 官 方 SDK 提供 示例项目	内置 AES 加 速器, 实测 AES 速率高 达 320Mbps[18]	608B 为专用加 密芯片, 面向 小报文/操作 级, 未标定持 续吞吐率	无片上 AES/PKA	片上 SAES 等
SHA-256	SAC 硬件支 持[20]	内置 SHA 加 速器	608B 支持 SHA 计算	需自行调 用开源库	Hash 模块 硬件支持
PMP 配置	数据手册 [43] 明确说 明支持 PMP, 官方 提供 SDK[20]	官方未见明 确说明	GD32VF103 理论支持 PMP, 但官方 文档未见明确 说明	官方手册 含 PMP 章节与寄 存器[21]	Arm 平台 无 PMP, 但具有类 似功能, 如 TrustZone
侧信道防护	无官方宣称	无官方宣称	608B 支持[17]	未公开	支持
开发难度	中等偏低: Nuclei Studio 工具 链, 官方 SDK	中等: SDK/ 社区较为活 跃	高: 该方案为 双板通信, I ² C 协议与状态机 协同较复杂	较低: 纯 软件实 现, 但性 能偏低	中等: 工 具链成 熟, 需熟 悉 SAES 等模块
其他	支持 TRNG	不支持 SM4	支持 TRNG	已停产	非 RISC-V

基于前述讨论，本项目希望能在中低端 MCU 开发平台上实践 AES、Ascon、SLH-DSA 等多种密码学算法，验证这些算法在资源受限环境下的效能；使用 PMP 配置相关寄存器的读写权限，保护存储密钥的敏感寄存器不会受到缓存溢出攻击；使用 TRNG 真随机数发生器和密码学算法对系统输入做掩码处理，实现简单的侧信道防护；使用 SLH-DSA 对固件做后量子数字签名，防止固件篡改。同时，考虑到研究背景是中低端 IoT 设备，价格、功耗等也是必须要考虑的因素。

本项目考虑了多种开发平台，这些开发平台的信息列在表格 1 中。基于多方面的因素及实际情况，本项目最终选择中国移动芯昇科技的 CM32M433R-START 开发板作为开发平台展开研究。CM32M433R 开发板采用 Nuclei N308 国产 RISC-V 核心，支持用户模式、PMP、TEE 等多个安全特性；内置密码算法硬件加速引擎（SAC），支持[43]AES、SHA、MD5、SM3、SM4 等算法加速，不仅能极大提高加解密速度，也带来额外的安全保障；内置 TRNG 熵源，生成硬件级真随机数，可为 AES 密钥、SLH-DSA 私钥盲化、侧信道防护掩码生成等多个步骤提供硬件支持。此外，芯昇科技开源了 CM32M433R 的软件开发包，包含相关驱动代码、样例代码及软硬件技术文档[20]，与 Nuclei Studio 工具链配合可实现一键部署示例工程，大大方便了开发者的学习与应用。图表 1 展示了 CM32M433R 的正面外观，图表 2 展示了 CM32M433R 的模块构成。



图表 1



图表 2

3.2 研究内容与研究方案

本项目基于 CM32M433R 开发板，围绕“轻量级 IoT 设备安全”的核心目标，构建“对称加密 + 后量子签名 + 侧信道防护”三位一体的方案，具体内容如下：

3.2.1 对称加密算法研究与技术对比

传统的加密算法，如 DES 和 AES，保证了处理对象的保密性（Secrecy），即保证了即使算法本身的细节是公开的，无密钥的第三方也无法轻易地从密文（Ciphertext）获取明文（Plaintext）。然而，这些算法本身并不提供对于处理对象的完整性（Integrity）和真实性（Authenticity）的支持，即在密文的传输过程中，不保证不会遭受到未授权的篡改，或篡改后能被迅速发现。为确保信息安全，认证加密（Authenticated Encryption, AE）是必要的。在三种常见的认证加密方法中，对密文生成认证码（Encrypt-then-MAC, EtM）是唯一可以达到认证加密安全性最高定义的方法[44]，前提是保证生成的认证码“强不可伪造”。例如，哈希函数 SHA-256 具有快速、确定性、难以分析、不可逆、无碰撞等特点，因此可用于生成认证码。

基于 AES-CTR 模式的 AES-GCM（Galois/Counter Mode，伽罗瓦计数器模式）和新兴的 Ascon 算法家族均属于带有关联数据的认证加密（Authenticated Encryption with Associated Data, AEAD）。AES-GCM[45]的加密部分与 AES-CTR 一

致，对每个块按顺序编号，再将编号与初始化向量组合，使用块密码算法（Block Cipher，这里即指 AES）加密，将结果与明文异或得到密文；而在认证部分，先由加密密钥生成认证子密钥，然后对密钥相关点做伽罗瓦有限域 $GF(2^{128})$ 上的乘法，其中每个密文块被认为是在该有限域上的一个本原多项式的系数，最后算得最终认证标签。而 Ascon 算法[33]的认证加密原理基于海绵函数（Sponge Function），将加解密与认证标签生成/验证耦合进一个 320 比特的海绵结构中，在实现上比 AES-GCM 更紧凑、更安全，还能提供一定程度的侧信道防护。

本项目将在开发板上集成 AES-GCM-SIV-128(硬件与软件实现)与 Ascon-128a 软件实现。与 AES-GCM 相比，AES-GCM-SIV[46][47]使用了基于伽罗瓦域乘法的合成初始化向量（Synthetic IV）来避免因不慎使用相同 Nonce 导致的密钥泄露。对于 AES-GCM-SIV，本项目使用 AES-ECB 进行密钥派生，正确分离加密密钥和认证密钥；使用标准定义的多重验证函数计算认证标签；使用 AES-CTR 进行加解密。对于 Ascon，本项目将实现纯软件的 `aead`、`permutations` 等关键函数。另外，本项目将支持大文件(KB级)加解密，通过自定义的通信协议，在电脑端使用 python 的 `pyserial` 库连接 MCU 的 UART 端口进行通讯；通过文件切分并妥善处理中断续传，实现在内存受限的 MCU 上进行大文件加解密。

CM32M433R 集成的 SAC 模块[43]可实现 AES-128 的硬件级加速，且官方 SDK 已给出 AES 示例工程代码。通过配置 RCC 使能 SAC 时钟，将 AES 设为 ECB 和 CTR 模式，配置好相关寄存器后即可使用。对于纯软件实现的 AES，是基于开源的 `tiny-AES-c` 库，使用 C 语言实现纯软件 AES，其性能完全取决于 CPU 的频率。基于纯软件实现的算法虽然在性能和空间占用上都必然劣于硬件实现的算法，但纯 C 语言的平台无关性使得该实现具有很强的可移植性。

对于 Ascon-128a 的实现，遗憾的是，根据 NIST 发布的标准化文件[33]，虽然 Ascon 算法的置换步骤采用哈希函数，但此处 Ascon 使用的是独特的 Ascon-p 哈希置换基结构，SAC 模块虽支持多种通用 SHA 算法，但均不能直接替代 Ascon-p 算法。因此，Ascon-128a 只能以纯软件形式应用，并利用直接内存访问（Direct Memory Access, DMA）、缓存优化、调度优化等策略做一定的吞吐量改善。具体地，基于官方开源的 `ascon-c` 库[23]将软件实现的 Ascon 算法移植入 SRAM，并配置 DMA 启用优化。最后，将硬件 AES、软件 AES 和 Ascon 的速率、资源占用、功耗等指标对比分析，量化 Ascon 在资源受限的环境下是否能代替 AES，并给出折中建议。

3.2.2 后量子签名技术应用与固件完整性检验

本项目将引入 SLH-DSA 后量子算法，并基于 SLH-DSA 数字签名加密固件、

Bootloader 签名验证来防止固件篡改（Firmware Tampering）攻击。为防止存储 SLH-DSA 私钥（以及对称算法的密钥）的寄存器被恶意读取，将敏感寄存器通过配置 PMP 阻止低特权级的访问，而公钥/验证基存入一次性可编程存储器（One-time Programmable, OTP）中。

根据 NIST 发布的 SLH-DSA 标准[35]，官方特别推荐采用 SHA-256 作为哈希算法，并推荐使用 SLH-DSA-SHA2-128s/128f 参数集，以更好地利用硬件支持。SHA-256 是一种非常流行的哈希算法，本开发板上的 SAC 也支持该算法的硬件加速。Saarinen 等人[36]的研究对硬件加速的 SLH-DSA 具有开创性的作用，他们开源了 SLoth 和从项目中提取出来的 SLH-DSA C 语言实现[24]，可移植入本平台并适配 SAC 模块的 SHA-256 算法。需要注意的是，虽然原论文实测得到两个数量级的提升，但那是基于团队自行设计的高性能专用加速器（SlotH）上，而本 SAC 的 SHA-256 尚无官方量化性能指标，尤其是 128f 参数集验证速度相对较慢，实际性能预估不会非常突出，目标耗时须以实测为准。

成功实现 SLH-DSA 后，即可实现固件篡改防护的主要流程。虽然 CM32M433R 并无自带的系统或 Bootloader，但官方 SDK 提供了通过 XMODEM 协议引导开发板进入 Bootloader 状态的方式，此状态下可装载用户自定义的二进制文件（即固件）。具体地，Bootloader 程序存储在 0x08000000 – 0x08010000 的寄存器地址段中，而用户程序存储在 0x08010000 – 0x08080000 段。用户首先将自己的完整代码工程编译成二进制文件（示例中为一个简单的亮灯程序），然后编译运行 Bootloader 主程序，使用 Tera Term 终端链接串口（USART1）与开发板交互。程序运行后，开发板将进入 Bootloader 态，可通过 log 串口打印选择菜单，然后输入 1 进入用户固件下载模式。在 Tera Term 下选择文件->传输->XMODEM->发送，选择之前编译好的二进制文件，等待发送完成后将再次进入菜单，此时选择 2 即可进入用户代码。

SLH-DSA 将在 Bootloader 验证阶段发挥作用。假设 Bootloader 镜像已烧录完毕（以上即为烧录过程），重新给开发板上电时（默认 BOOT0, BOOT1 两个跳线接口已正确设置[43]），Bootloader 就会开始工作。从状态机的角度出发，Bootloader 在上电后先进入最小初始化（或复位）状态，然后读取 PA0_WKUP 按钮的状态。若该按钮为按下状态（类似于电脑启动时按住特定按钮进入 BIOS），则系统主动进入 IAP（启动菜单）态；否则，进入验证态。在验证态，Bootloader 程序将读取并加载镜像头，对完整的固件镜像计算哈希值（使用 SCA 硬件加速），读取 OTP 中存储的公钥，最后调用 SLH-DSA 签名验证算法，结合固件哈希值和公钥进行验证。若验证通过，则进入跳转态，系统将进入用户程序寄存器段，Bootloader 状态正式结束；否则，进入 IAP 态，此状态类似电脑的 BIOS/UEFI 系统，用户若持有

正确的固件文件，可输入 1 进入 XMODEM 下载模式，重新烧录固件，然后输入 2 重新进入验证态，也可手动触发 IAP 态来验证流程的完整性。

以上流程虽然忽略了很多实施细节，例如长时间验证过程中可能需要暂时停用看门狗（Watchdog）避免误复位、跳转态时最后一次验证或设置向量表等，但应该足以展示 Bootloader 的基本理念，体现基于信任根的安全启动（Secure Boot）逻辑。引入 SLH-DISA 作为核心数字签名验证算法后，虽然可能会延长 Bootloader 跳转至用户态的时间，但大幅增强了系统安全启动的保障性，并引入了抗量子计算攻击的安全特性。

3.2.3 侧信道防护优化

侧信道攻击种类繁多、防护难度较大，限于个人能力与项目条件等原因，本项目仅讨论简单的基于掩码的功耗分析保护。

高熵值的掩码来源是 CM32M433R 集成的真随机数发生器（True Random Number Generator, TRNG）。NIST 在 2018 年[25]给出了一个推荐的 TRNG 设计标准。本开发板集成的 TRNG[43]由模拟随机源（产生较高速率的随机数比特流）和数字后处理电路（提高随机数质量）组成，官方文档中有详细的 TRNG 使用步骤，官方 SDK 也有 TRNG 的使用样例，这里不再赘述。

使用 TRNG 生成随机掩码后，可应用到上述多种算法中，提高侧信道防护。例如，将掩码与 AES 明文和密钥做异或、与 Ascon 的每个核心寄存器独立做异或，使得信息在传递时不会泄露功耗侧信道信息；将 SLH-DISA 的私钥种子与掩码做异或，此过程称为盲化（Blinding），由于每次掩码都不同，攻击者即使采样上千万次签名轨迹，也无法叠加分析出其统计特征，从而使侧信道攻击失效。

在上述过程中，可使用 DMA 将掩码预载入缓冲区，以减少 CPU 主动操作造成的时间侧信道泄露；使用 PMP 阻止非法读取掩码，实现存储安全。构建完成后，使用测试向量泄露评估（Test Vector Leakage Assessment, TVLA）[26]分别对“有/无掩码”的版本功耗轨迹做 t-test。一般地，若 $|t| < 4.5$ ，可认为防护有效。

4 进度安排及预期目标

4.1 进度安排

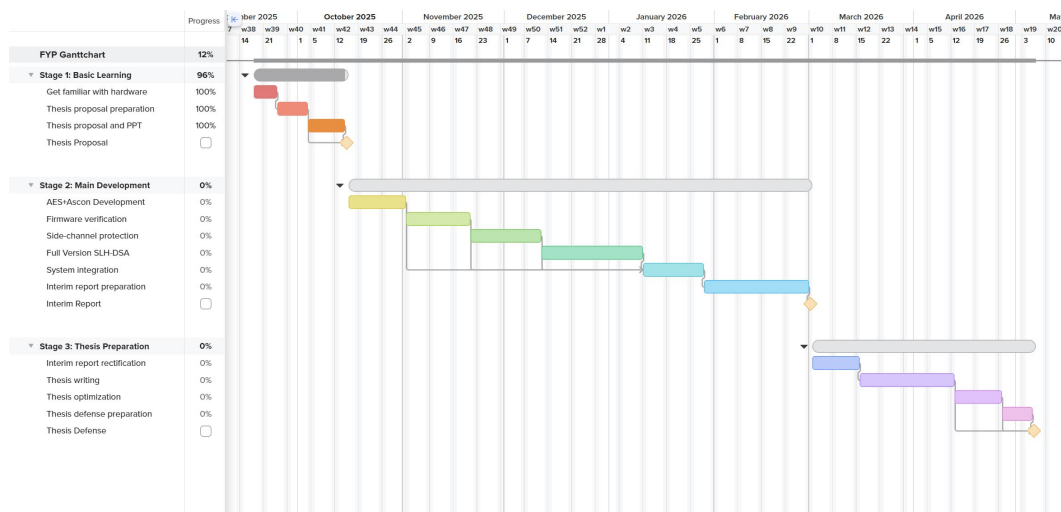
本项目的进度安排如表格 2 所示。

表格 2

日期范围	核心任务	阶段性成果
9.19 – 9.24	安装 Nuclei Studio, 完成点灯 使能 UART 调试 标注关键寄存器地址	硬件测试截图 寄存器地址标注笔记
9.25 – 10.3	文献调研与补充 开题报告撰写	文献清单 开题报告部分内容
10.4 – 10.15	开题报告完善 制作开题答辩 PPT	开题报告终稿 答辩 PPT
10.16 – 11.1	对称加密算法开发: 基于 SAC 实现硬件 AES, 基于 tiny-AES-c 库实现软件 AES, 基于 ascon-c 实现软件 Ascon	AES/Ascon 代码 速率表 TRNG 熵值报告 (健康监测)
11.2 – 11.20	固件校验: 验证 Bootloader 完整流程 SLH-DSA 密钥生成: 将私钥置于 PMP 保护段, 公钥置于 OTP	固件校验代码 SLH-DSA 密码对 固件篡改测试视频
11.21 – 12.11	侧信道防护: TRNG 生成掩码, 与 AES、Ascon、SLH-DSA 相关部分异或, 使用 TVLA 验证	掩码生成/异或/盲化代码 侧信道报告 (TVLA 测试结果, 示波器功耗波动监测)
12.12 – 1.10	SLH-DSA 完整实现: 基于推荐参数集和 SAC 硬件哈希加速实现签名算	SLH-DSA 签名/验证代码 固件篡改攻击与性能测试

	法，在 Bootloader 中集成 签名验证逻辑	
1.11 – 1.28	系统集成： 统一接口：（如 mode=0:AES/1:Ascon, sign_mode=SLH-DSA） 安全启动链路：安全启动， SLH-DSA 验证->FLASH 校验/哈希->AES/Ascon 常 规加密任务 自动脚本进行 10 万次压 力测试，监测关键寄存器 的稳定性	第一版完整系统固件 压力测试报告 ROM/SRAM 占用资源统 计
1.29 – 2.28	中期报告准备： 整理文档 制作演示视频	中期报告 演示视频
3.1 – 3.15	中期答辩与整改： 优化 SLH-DSA：补充私钥 PMP 保护 优化 Ascon：动态掩码更 新（每轮加密可选择刷新 掩码值，提高安全性） 补充负面测试：畸形数据 输入，监测错误标志位	答辩反馈记录表 PMP 配置代码 负面测试结果
3.16 – 4.12	论文核心撰写	论文初稿 性能对比表
4.13 – 4.26	论文优化	论文终稿
4.27 – 5.5	答辩准备	答辩 PPT 完整的项目文件包（代码+ 文档+视频）

本项目的甘特图如图表 3 所示。



图表 3

4.2 预期目标

考虑到 CM32M433R 的实际性能、软件方案的限制、实验室的条件限制（5.1 等多方面因素，表格 3 给出了多个评估维度下的预期目标。本项目承认，评估预期指标有很大的不确定性，若在之后的研究过程中有所变动，相关目标可能无法达到。

表格 3

指标名称	测试方法	目标值/预期范围
AES-128 硬件加密吞吐率	在 DUT 上运行大块数据加解密循环，利用 MCU 内部计时或用示波器（GPIO 触发）测单次/连续吞吐，记录 CPU/外设状态以消除 I/O 瓶颈，DM858 用于平均电流记录（能耗验证）	≥ 80 Kbps
Ascon-128a 软件加密吞吐率	用 GPIO 触发 + 示波器测单次耗时，DMA 辅助时用示波器或软件计数器对比有/无 DMA 的时延	≥ 8 Kbps
AES 掩码实现性能功耗	对比无掩码与有掩码下的单次耗时（GPIO trigger +	AES 掩码开销 $\leq 10\%$ ，Ascon 掩码 $\leq 15\%$

	scope), 取多次平均	
加密能耗效率	在分流 ($0.1 - 1 \Omega$) 上用 DHO814 捕获 $I(t)$, 与 V_{cc} (或用稳压的 GPD 电压) 结合积分计算能量, DM858 用作长期平均参考, 但需要精密分流与差分放大器或差分探头	$\leq 0.30 \text{ mJ / KB}$
签名生成耗时	在软件中在签名函数入口 / 出口拉触发 GPIO, DHO814 测时基, 多次运行取均值与方差	$\leq 50 \text{ ms}$
签名验证耗时	同上	$\leq 30 \text{ ms}$
引入盲化的时间开销	对比同一签名实现有/无盲化的耗时差异, 用示波器或软件计时器记录	$< 10\%$ 时间开销
私钥派生耗时	同上, 用 GPIO 或内部计时器记录	$\leq 40 \text{ ms}$
签名大小	直接测输出签名字节数	$\leq 10 \text{ KB}$
功耗泄露显著性 (TVLA)	参考相关论文[27]的方法论, 采集 $N_A = N_B = 5000$ traces 初测, 每两组数据独立采集两次 TVLA	$t < 4.5$ ($N \geq 10^4 \text{ traces}$)
DPA 攻击成功率	收集大量 traces ($10k - 50k$), 用 CPA 工具 (如 ChipWhisperer 软件、Python 实现) 尝试密钥恢复, 但需高质量 traces (差分探头/放大器)	$50k \text{ traces 内} \leq 2\%$
安全启动完整性验证耗时	在启动序列插入触发点并用示波器测量, 配合日志记录确保每次启动动作完	$\leq 75 \text{ ms}$

整		
固件更新验证成功率	自动化脚本重复 OTA 并验证签名, DM858 可用于监测供电异常	$\geq 98\%$ (100 次测试)
加密任务 CPU 占用率	在 RTOS 中用系统监控 (任务运行时间统计) 或用示波器测 GPIO 并对比空闲时间计算 CPU 占比	$\leq 40\%$
活跃态平均功耗	用分流+示波器积分法或用 GPD+DM858 做长期平均对比	$\leq 50 \text{ mW}$ (连续运行)
长时间稳定运行	自动化脚本循环执行, DM858 记录平均值, 示波器间断抓拍关键轨迹	24 h 连续运行无异常

5 已具备和所需的条件和经费

5.1 实验室条件和经费保障

在实验室条件方面，香港城市大学青岛研究院（CityUHKQRI）实验室目前已具有如下设备：GPD-4303S 直流电源、RIGOL DM858 数字万用表、RIGOL DHO814 数字示波器、ATTEN 焊台等。实验室目前暂未开放，但也不断有新设备入驻。

经费方面，本项目目前已购置 CM32M433R 开发板一块和若干杜邦线。CM32M433R 已在板上集成 USB 转串口模块和调试芯片，无需额外购置 JTAG 调试器（当然也保留了原始接口，可连接至 J-Link、RV-Link 等调试器）。CityUHKQRI 承诺提供一定的经费支持，日后若有学术需求可申请使用。此外，哈尔滨工业大学（深圳）已出资订阅多个学术网站，可免费下载论文；各类网上资料、开源项目和代码均无需额外费用。

5.2 所需条件和经费

基于前述期望目标，可能需要高精度的用于功耗测量的低阻分流电阻、差分电流探头等设备，如有可能，也可额外购置一块备用的 CM32M433R 开发板，以同时测试不同的加密方案或作为紧急情况下的后备保障。不过在可预见的开发周期，本项目暂无急需的实验室条件或经费。

6 预计困难及解决方案

前述文段已提到部分困难（3.2 ， 4.2 ），以下将分点总结。

6.1 CM32M433R 官方文档部分缺失、信息过时

本项目的核心之一就是板上集成的 SAC 模块。然而，官方文档[43]对 SAC 的描述仅有一页，并提到“密码算法性能及使用请联系中移物联销售人员”。截至目前，笔者已与中移物联销售人员取得联系，但从沟通结果上看，相关人员似乎并无“密码算法性能及使用”的文档。

因此，官方提供的与 SAC 模块有关的资料仅剩 SDK 工程文件。虽然在使用时，这些资料信息已经足够，但若想查看 SAC 的底层加密算法实现，就会发现相关库函数已被编译成静态链接库文件（据查看，所有 AES、DES、SHA、SM4 底层函数均已被编译成.a 文件），只能调用而无法看到源代码实现，推测芯昇科技在这里采用保密的政策。同样地，由于官方未提供密码算法性能，本文在拟定 CM32M433R 的加密算法性能指标时仅考虑了 CPU 主频(144MHz)，而未考虑 SAC 的实际性能。基于以上原因，本项目在硬件加速加密算法的底层实现上将有所缺失，相关指标的实现存在不确定性。

另外在实践中发现，官方提供的示例工程无法直接编译通过，原因是官方的设置已经部分过时。目前的解决方案是替换编译工具链为 riscv64-unknown-elf、删去过时的 P 扩展、将优化级别改为-O0（无优化），重新编译再运行即可。目前已成功用该方案实现亮灯、USART 通讯、按键与外部中断等基础功能。

6.2 SAC 模块配置复杂，可能会初始化失败

由于寄存器的时序配置要求严格，且文档可能存在细节遗漏，不保证 SAC 能顺利配置成功。目前的解决方案是分步调试，调试过程中持续检查相关寄存器状态，再基于已有信息分析问题源头。

6.3 硬件资源受限

由于 CM32M433R 自身的计算能力、存储空间等限制，高效、低功耗地实现加密算法并非易事，尤其是这些算法的基本原理注定需要频繁的内存访问和计算，导致平台资源占用率居高不下，影响设备整体性能。DMA 若能成功配置，将有效缓解上述问题，因为它可分担 CPU 搬运数据的工作，减少 CPU 的占用；SAC 硬件加速也是减少计算开销的重要部件。若问题仍存在，将考虑降低加密算法的精

度和性能来换取更高的能耗比表现，采用灵活的配置策略，平衡安全和性能。

6.4 指标测定方法过于理想化，实验室条件不足

表格 3 中提到的测试方法仅是基于设备的理论功能的理想化分析，实际上由于多方面因素和实验的不确定性，这些方法不一定成立，尤其是侧信道防护验证要求高精度的功耗测量。对此，本项目计划先完成功能性验证，即各个子模块的正确性，然后才进入实测环节。若条件不允许，计划将实测指标调整为定性分析，引用权威文献结论辅助论证侧信道防护的有效性，但不能因为该问题拖慢后续研发进度。

6.5 SLH-DSA 性能低，需求计算资源较多

作为 RSA 的替代者，SLH-DSA 也部分继承了 RSA 的缺点：运行速度较慢、占用资源较多、生成的签名可能过长等等。SLH-DSA 尚处于不断优化的阶段，其性能瓶颈在资源受限的平台上尤为突出。此外，SAC 对 SLH-DSA 的支持并不完美，利用硬件加速 SHA-256 来提升 SLH-DSA 速度可能存在工程上的困难。

对此，本项目将积极探讨可能的优化方案，如优化哈希计算过程、利用 DMA 加速计算结果和中间值的存取。若已用完所有的优化方案，而 SLH-DSA 性能仍然过差，本项目将如实记录实测数据，并考虑替换成简单的 SHA 签名与认证。

6.6 系统集成的潜在困难

在统一各模块接口时，可能会遇到寄存器冲突、PMP 配置错误、内存数据污染等潜在问题。解决问题的最好方案就是严格对照官方文档的寄存器地址，先规划好各模块的内存分区，再在调试时严格检查寄存器状态，定位问题所在。若问题实在无法解决，将考虑放弃系统集成，改为在两块 CM32M433R 上独立验证子模块的功能性。

除以上问题之外，也存在不少的非工程问题，例如笔者的能力不足、知识体系构建不完整、项目开发时间安排紧张等。对此，并无很好的解决方案，唯有自律精进、充分发挥主观能动性、主动查阅并学习各类公开资料、利用甘特图实现自我时间管理等最佳实践才是良方。

7 参考文献

- [1] LIONEL S. Number of IoT connections worldwide 2022-2034 [DB/OL] . (2025-6-26) [2025-10-7] . <https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/>
- [2] 劳婵绚. CVA6项目中PMP条目数量限制问题的分析与解决 [EB/OL] . (2025-7-1) [2025-10-7] . <https://blog.gitcode.com/fff1e1d96b814966336eea1d325e5e52.html>
- [3] HP Inc. HP t430 and t638 Thin Clients - Firmware Tampering Vulnerability [DB/OL] . (2024-11-21) [2025-10-7] . <https://nvd.nist.gov/vuln/detail/CVE-2023-5409>
- [4] 胡伟, 张家琦, 等. 首个国内自主挖掘的RISC-V处理器设计中危漏洞研究 [EB/OL] . (2024-4-24) [2025-10-7] . <https://cn-sec.com/archives/2687427.html>
- [5] 闫润, 黄立波, 成元虎, 等. RISC-V特权架构配置的硬件实现影响研究[J]. 小型微型计算机系统, 2024, 45(04): 1018-1024. DOI: 10.20009/j.cnki.21-1106/TP.2022-0566.
- [6] 王杰, 王鹏. 面向RISC-V平台的安全高效固件可信平台模块设计与实现[J]. 电子与信息学报, 2025, 47(07): 2385-2395.
- [7] 秦体红, 汪宗斌, 刘洋, 等. 基于商密SM9算法同态加密方案[J]. 信息安全研究, 2024, 10(06): 513-518. DOI: CNKI: SUN: XAQY. 0. 2024-06-004.
- [8] 杨万. RISC-V处理器核中AES加密的硬件设计和指令扩展[D]. 杭州电子科技大学, 2024. DOI: 10.27075/d.cnki.ghzdc.2024.000576.
- [9] 环球网. 安全团队演示首个基于浏览器的旁道攻击 [EB/OL] . (2021-3-12) [2025-10-7] . <https://www.toutiao.com/article/6938560887166861838/>
- [10] M. R. FADIHEH, D. STOFFEL, C. BARRETT, S. MITRA, AND W. KUN Z. “Processor Hardware Security Vulnerabilities and their Detection by Unique Program Execution Checking”. 2019 Design, Automation Test in Europe Conference Exhibition (DATE). 2019, pp. 994–999.
- [11] DEJUN XU, KAI WANG, JING TIAN. A Hardware-Friendly Shuffling Countermeasure Against Side-Channel Attacks for Kyber[J]. arXiv:2407.02452
- [12] BEN GRAS, KAVEH RAZAVI, HERBERT BOS, AND CRISTIANO GIUFFRIDA. “Translation leak-aside buffer: Defeating cache sidechannel protections with {TLB} attacks”. 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018, pp. 955–972.
- [13] PETER W. SHOR. Polynomial-Time Algorithms for Prime Factorization and

- Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing. 1997, 26 (5): 1484–1509. Bibcode:1995quant.ph..8027S. arXiv:quant-ph/9508027 doi:10.1137/S0097539795293172.
- [14] 吴晓杰.后量子数字签名SLH-DSA算法硬件设计与实现[D].华中科技大学,2024. DOI:10.27157/d.cnki.ghzku.2024.002570.
- [15] ABRAHAM GONZALEZ, BEN KORPAN, JERRY ZHAO, ED YOUNIS, AND KRSTE ASANOVIĆ. “Replicating and Mitigating Spectre Attacks on an Open Source RISC-V Microarchitecture”. Third Workshop on Computer Architecture Research with RISC-V (CARRV 2019), Phoenix, AZ, USA. 2019.
- [16] ELKE DE MULDER, SAMATHA GUMMALLA, AND MICHAEL HUTTER. “Protecting RISC-V against side-channel attacks”. 2019 56th ACM/IEEE Design Automation Conference (DAC). IEEE. 2019, pp. 1–4.
- [17] Microchip Technology Inc. AVR-IoT Cellular Mini Hardware User Guide [CP/OL]. [2025-10-8]. <https://onlinedocs.microchip.com/oxy/GUID-EA329A0D-DD6D-43F1-9B88-C43325885A4D-en-US-2/GUID-7BC4FB6F-D18E-4C40-A047-BA2A8D8FFB95.html>
- [18] WILLIAM. K210开发板高级加密加速器实战指南 [EB/OL]. (2025-5-21) [2025-10-8]. <https://bbs.huaweicloud.com/blogs/452795>
- [19] STMicroelectronics. STM32U5 series - PDF Documentation [DB/OL]. [2025-10-8]. <https://www.st.com/en/microcontrollers-microprocessors/stm32u5-series/documentation.html>
- [20] XinSheng Technology Inc. CMIOT.CM32M4xxR_Library [CP/OL]. (2023-6-15) [2025-10-9]. https://github.com/CMIOT-XinShengTech/CMIOT.CM32M4xxR_Library/tree/main/Docs
- [21] SiFive, Inc. SiFive FE310-G002 Manual [DB/OL]. (2021-3-25) [2025-10-9]. <https://starfivetech.com/uploads/fe310-g002-manual-v1p0.pdf>
- [22] KOKKE. tiny-AES-c [CP/OL]. (2024-8-4) [2025-10-10]. <https://github.com/kokke/tiny-AES-c>
- [23] MARTIN SCHLÄFFER et al. ascon-c [CP/OL]. (2025-1-25) [2025-10-10]. <https://github.com/ascon/ascon-c>
- [24] MARKKU-JUHANI O. SAARINEN. slh-dsa [CP/OL]. (2025-6-15) [2025-10-10]. <https://github.com/slh-dsa>
- [25] TURAN, M.S., BARKER, E., KELSEY, J., MCKAY, K.A., BAISH, M.L., BOYLE, M. Recommendation for the entropy sources used for random bit generation. [S] NIST Special Publication SP 800-90B (2018). DOI 10.6028/NIST.SP.800-90B
- [26] COOPER J, DEMULDER E, GOODWILL G, et al. Test Vector Leakage As

- essment (TVLA) methodology in practice[C]. International Cryptographic Module Conference, Shanghai, China, 2013.
- [27] EISENBARTH T, CHAPMAN J, VALME R. Test Vector Leakage Assessment Development[J]. [2025-10-11]
- [28] ANDREW WATERMAN, KRSTE ASANOVIĆ, SiFive Inc. The RISC-V Instruction Set Manual Volume II: Privileged Architecture[S]. (2019-6-8) [2025-10-12]
- [29] CHEANG K, RASMUSSEN C, LEE D, et al. Verifying RISC-V Physical Memory Protection[A/OL]. arXiv, 2022[2025-10-05]. <http://arxiv.org/abs/2211.02179>. DOI:10.48550/arXiv.2211.02179.
- [30] DONG R, CUI B, SUN Y, et al. A combined side-channel and transient execution attack scheme on RISC-V processors[J/OL]. Computers & Security, 2025, 150: 104297. DOI:10.1016/j.cose.2024.104297.
- [31] 黎明热忱. 简单了解一下RVA23 Profile架构规范体系 [EB/OL] (2025-7-28) [2025-10-13] <https://bbs.21ic.com/icview-3474180-1-1.html>
- [32] DOBRAUNIG C, EICHLSEDER M, MENDEL F, et al. Ascon v1.2: Lightweight Authenticated Encryption and Hashing[J/OL]. Journal of Cryptology, 2021, 34(3): 33. DOI:10.1007/s00145-021-09398-9.
- [33] SONMEZ TURAN M. Ascon-Based Lightweight Cryptography Standards for Constrained Devices: NIST SP 800-232[R/OL]. Gaithersburg, MD: National Institute of Standards and Technology, 2025: NIST SP 800-232[2025-10-12]. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-232.pdf>. DOI:10.6028/NIST.SP.800-232.
- [34] LU T. A Survey on RISC-V Security: Hardware and Architecture[A/OL]. arXiv, 2021[2025-09-16]. <http://arxiv.org/abs/2107.04175>. DOI:10.48550/arXiv.2107.04175.
- [35] National Institute of Standards and Technology (US). Stateless hash-based digital signature standard: NIST FIPS 205[R/OL]. Washington, D.C.: National Institute of Standards and Technology (U.S.), 2024: NIST FIPS 205[2025-10-14]. <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf>. DOI:10.6028/NIST.FIPS.205.
- [36] SAARINEN M J O. Accelerating SLH-DSA by Two Orders of Magnitude with a Single Hash Unit[C/OL]//REYZIN L, STEBILA D. Advances in Cryptology – CRYPTO 2024. Cham: Springer Nature Switzerland, 2024: 276-304. DOI:10.1007/978-3-031-68376-3_9.
- [37] BEN MARSHALL (EDITOR), MARKKU-JUHANI O. SAARINEN, NATHAN MENHORN et al. RISC-V Cryptography Extensions Volume I[S]. (2022-2-18) [2025-10-13]
- [38] TAIN C, PATIL S, AL-ASAAD H. Survey of Verification of RISC-V Processors[J/OL]. Journal of Electronic Testing, 2025, 41(2): 111-138. DOI:10.1007/s10836-025-06169-3.
- [39] BOVE D. Secure Services for Standard RISC-V Architectures[M/OL]//Proceedings of the 17th International Conference on Availability, Reliability and

- d Security. 2022: 1-7[2025-09-11]. <https://dl.acm.org/doi/10.1145/3538969.3538998>. DOI:10.1145/3538969.3538998.
- [40] BOUBAKRI M, CHIATANTE F, ZOUARI B. Towards a firmware TPM on RISC-V[C/OL]//2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). 2021: 647-650[2025-09-11]. <https://ieeexplore.ieee.org/document/9474152/>. DOI:10.23919/DATE51398.2021.9474152.
- [41] LAURENT SAUVAGE, SOFIANE TAKARABT, AND YOUSSEF SOUIS SI. “Secure silicon: Towards virtual prototyping”. In: 2017 International Symposium on Electromagnetic Compatibility-EMC EUROPE. IEEE. 2017, p p. 1–5.
- [42] FRITZMANN T, SIGL G, SEPÚLVEDA J. RISQ-V: Tightly coupled RISC-V accelerators for post-quantum cryptography[J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020: 239-280.
- [43] 中移物联网有限公司. CM32M4xxR 系列32 位RISC-V 微控制器用户手册 V1.4 [M]. (2022-5-1) [2025-10-13]
- [44] ISO/IEC 19772:2009 - Information technology -- Security techniques -- Authenticated encryption[EB/OL]. (2016-04-16)[2025-11-26]. https://web.archive.org/web/20160416165453/http://www.iso.org/iso/catalogue_detail.htm?csnumber=46345.
- [45] Dworkin M J. Sp 800-38d. recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac[M]. National Institute of Standards & Technology, 2007.
- [46] GUERON S, LINDELL Y. GCM-SIV: Full Nonce Misuse-Resistant Authenticated Encryption at Under One Cycle per Byte[A/OL]. Cryptology ePrint Archive, 2015[2025-11-25]. <https://eprint.iacr.org/2015/102>.
- [47] GUERON S, LINDELL Y. Better Bounds for Block Cipher Modes of Operation via Nonce-Based Key Derivation[A/OL]. Cryptology ePrint Archive, 2017[2025-11-26]. <https://eprint.iacr.org/2017/702>.