# 周报 向嘉豪(2025-10-13)

**摘要:** 本周完成 ML-DSA 论文实验方法论体系构建与实验环境部署准备。完成第 4 节实验方法论框架设计，建立 ARM Cortex-M4 性能评估体系。实验平台采用 STM32F407VG(168MHz、1MB Flash、192KB SRAM)配合 pqm4 库实现 ML-DSA 测试。测量框架利用 DWT 单元实现周期级计时与栈水印内存剖析。基准测试覆盖密钥生成、签名、验证全流程，包含 10/50/100 字节 IoT 载荷，建立 ECDSA P-256 对比基线。评估指标涵盖计算性能、内存占用、协议开销三维度。实验环境已完成 STM32 配置与工具链部署，pqm4 移植与 MQTT 集成进行中。

**下周计划:** 1) 完成 pqm4 库 ARM Cortex-M4 平台移植与功能验证。 2) 启动 ML-DSA 基准测试数据采集，完成密钥生成、签名、验证性能基线测量。 3) 实现 MQTT 载荷嵌入式签名原型系统，测量协议级传输开销。

## 1 ML-DSA 论文第 4 节实验方法论体系完成

### 1.1 实验平台架构设计

硬件平台选型 本周完成实验平台架构设计，采用 ARM Cortex-M4 微控制器作为评估平台，代表工业监控、智慧农业等中端 IoT 设备典型配置。硬件选型为 STM32F407VG 开发板，搭载 168MHz 主频 ARM Cortex-M4F 内核，配备 1MB Flash 与 192KB SRAM，真实反映需要密码学认证的资源约束环境。软件采用 ARM GCC 10.3.1 工具链，启用-O3 优化以实现最佳性能。

密码学库集成 ML-DSA 实现源自 pqm4 参考库，提供 ARM Cortex-M4 优化的全部三个参数集(ML-DSA-44/65/87)并通过 NIST 测试向量验证。集成 micro-ecc 库实现 ECDSA P-256 作为对比基准，采用相同编译优化确保可比性。MQTT 集成采用 Eclipse Paho 嵌入式 C 客户端，配置 QoS 1 与 5 秒保活间隔，通信目标为 Mosquitto 2.0.15 代理。网络连接利用 ESP32-WROOM-32 模块提供 WiFi 接入。

### 1.2 性能测量框架建立

计算性能测量 利用 ARM Cortex-M4 DWT 硬件单元通过 DWT_CYCCNT 寄存器提供周期精度计时。该硬件方法消除软件剖析开销，实现单周期分辨率。测量涵盖完整操作流程，包括预处理、核心计算、结果格式化阶段。

内存占用剖析 采用静态与动态测量相结合的方法。静态消耗通过 arm-none-eabi-size 从编译输出提取，量化代码与数据段 Flash 需求。动态剖析采用栈水印技术，以 0xDEADBEEF 哨兵值初始化栈区域，执行后扫描确定峰值利用。能耗评估采用 INA219 传感器以 100Hz 频率测量供电电流，支持电池供电设备功率分析。

### 1.3 基准测试方案设计

密码学操作评估 基准测试涵盖全部 ML-DSA 操作的系统性评估。密钥生成测量完整公私钥对生成，签名生成捕获端到端延迟包括拒绝采样迭代，验证测量签名验证计算成本。每个参数集横跨 10 字节传感器读数、50 字节遥测包、100 字节状态报告三类 IoT 载荷，捕获真实通信模式。ECDSA P-256 基线执行相同操作序列，全部测量在相同环境条件下执行。

统计严谨性 每项操作执行 1000 次迭代并消除离群值。报告中位数执行时间提供稳健估计，配合四分位距量化变异性。对签名生成额外报告最小与最大执行时间以刻画性能边界。

### 1.4 MQTT 集成测试协议

端到端认证工作流 MQTT 集成评估载荷内嵌入 ML-DSA 签名的认证工作流。发布者生成签名并附加到载荷，通过 MQTT PUBLISH 传输复合消息。订阅者接收消息，提取签名，检索公钥并验证真实性。端到端延迟测量涵盖签名生成、消息传输、网络传播、代理路由、接收反序列化、签名验证完整生命周期。测试评估 QoS 0/1/2 三个级别的开销特征。

### 1.5 评估指标体系

多维评估框架 指标涵盖计算性能(CPU 周期、执行时间、操作速率)、内存占用(代码尺寸、静态 RAM、栈消耗)、协议影响(消息尺寸开销、传输延迟、验证延迟)三大维度。比较分析采用开销比率 R = M{ML-DSA} / M{ECDSA}量化后量子迁移成本，为资源受限 IoT 环境的部署规划提供洞察。

## 2 实验环境搭建进展

### 2.1 STM32 开发板平台配置

硬件验证 已获取 STM32F407VG 开发板并完成硬件验证。通过 ST-Link 建立稳定主机-目标板通信，确认 USB 供电与串口通信正常。配置 ARM GCC 10.3.1 工具链与 OpenOCD 调试服务器，实现程序烧录、断点调试功能。配置 115200 波特率串口终端用于测试输出与数据采集。

### 2.2 pqm4 密码学库移植

项目结构分析 从 GitHub 获取 pqm4 源码，完成项目结构分析，识别核心模块包括 poly.c、ntt.c、sign.c、randombytes.c 等组件。库采用模块化设计，支持通过编译标志选择不同参数集。

移植适配 正在进行 STM32 平台移植适配。主要挑战包括硬件抽象层适配、RNG 外设对接、DWT 计数器配置。需整合 pqm4 Makefile 与 STM32 HAL 构建脚本。已完成基础编译环境配置，正解决头文件路径与链接器脚本适配。

### 2.3 MQTT 客户端集成

Paho MQTT 库 评估 Eclipse Paho 嵌入式 C 客户端集成可行性。该库提供轻量级 MQTT 3.1.1 实现，符合资源受限要求。识别关键配置参数包括最大消息尺寸(容纳 2.4-4.6KB 签名载荷)、网络缓冲区、QoS 支持。模块化网络接口支持对接 ESP32 WiFi 模块。

ESP32 对接方案 规划 ESP32-WROOM-32 与 STM32 的 UART 串口 AT 指令通信方案。ESP32 处理网络协议栈，STM32 专注密码学运算与 MQTT 逻辑，符合性能测量隔离原则。正准备固件烧录与 AT 接口验证。

### 2.4 进度与挑战

实验环境已完成硬件验证、工具链配置、库源码分析等基础工作。当前核心任务为 pqm4 平台移植与 MQTT 集成，预计 1-2 周完成。待解决挑战包括 RNG 外设对接确保密码学随机性、DWT 计数器精确配置验证测量准确性、MQTT 签名消息格式设计。这些问题是下周核心目标，直接影响实验数据采集阶段。