

# 周报 向嘉豪(2025-10-27)

**摘要:** 本周完成第二篇论文 R2 小修全部修订任务并成功提交。完成[安全性分析章节\(Point 1\)](#), 在第 IV-C3 节新增“Security Considerations”子章节, 阐述位切片恒定时间执行特性、缓存时序攻击防护能力、优化技术对安全属性保持以及门级掩码优势, 添加两篇侧信道攻击相关文献。完成[图表质量提升\(Point 4\)](#), 优化 Table 1 格式增强可读性, 将全部图表转换为 DrawIO 矢量图并提升字体大小至 45px。完成审稿意见回复文档全部 Response 部分撰写与修订稿件最终校对。2025 年 10 月 26 日成功提交 R2 修订版本。

下周计划: 1) 推进第四篇论文写作与实验

## 1 安全性分析章节完成(Point 1)

### 1.1 Security Considerations 新增

**章节定位** 在第 IV-C 节 “Implementation Results” 末尾新增子章节 “Security Considerations” (Sec. IV-C3), 作为实现结果分析的安全性补充说明。该章节回应审稿人关于侧信道防护能力的关切, 阐明位切片实现的安全优势与局限性。

**恒定时间执行特性** 阐述位切片实现固有的恒定时间与恒定流特性, 说明操作均匀分布于所有位而无数据依赖分支或内存访问模式。该特性提供针对缓存时序攻击的内在防护能力, 攻击者无法通过微架构泄漏获取密钥相关信息。引用文献[25] Aldaya et al., TCHES 2019 关于 RSA 密钥生成缓存时序攻击研究, 强化论述可信度。

**优化技术安全属性保持** 明确说明 OPO 算法(线性层置换优化)与增强 BGC 编码(S 盒优化)保持恒定时间特性, 通过维持逐位并行操作而不引入条件执行路径或密钥依赖表查找。该论述回应审稿人对“性能优化是否损害安全性”的潜在疑虑, 证明优化与安全性兼容。

**门级掩码优势** 强调位切片实现的独特优势: 门级掩码对策可直接应用于保护基本布尔运算, 同时保留原始操作序列与性能优化。引用文献[26] Balasch et al., CHES 2015 关于 DPA、位切片与掩码技术研究, 说明该方法相比查找表实现所需算法级掩码更系统化且开销更低。

**局限性与部署建议** 明确指出恒定时间执行不防护功耗分析或电磁侧信道攻击, 需在安全关键环境部署额外针对性对策。该声明体现学术严谨性, 避免过度宣称安全属性。

## 2 图表质量全面提升(Point 4)

### 2.1 Table 1 格式优化

**可读性增强** 改进 Table 1 “Mathematical Notation for Bitsliced Implementation” 排版, 通过三项技术措施提升可读性: (1) 增加行间距避免内容拥挤; (2) 采用左对齐描述列增强视觉舒适度; (3) 扩展列间水平间距至 2em 消除视觉混淆。修改后表格符号与描述对应关系更清晰, 便于读者快速检索符号定义。

## 2.2 全部图表矢量化

**DrawIO 矢量图转换** 将全部 5 幅图表(Fig. 1-5)使用 DrawIO 重新绘制并导出为 PDF 矢量格式, 确保任意缩放下保持清晰度。矢量格式替代栅格图像满足 IEEE 出版质量标准, 解决审稿人“图表难以阅读”问题。

**字体大小提升** 统一将全部图表字体大小从 35px 提升至 45px(28.6% 增幅), 显著增强打印与投影场景可读性。字体大小调整遵循 IEEE 会刊图表规范, 确保 6 英寸宽度打印条件下文字清晰可辨。

**冗余元素清理** 优化画布尺寸移除不必要空白区域, 提升图表信息密度。Fig. 5 特别移除重复位置框, 消除视觉冗余, 强化核心置换变换逻辑表达。

## 3 审稿意见回复文档完成

### 3.1 Response 全面撰写

**Point 完整回复** 完成 reviewer/response.typ 全部 Response 部分撰写, 包括:

- Editor Comments: 确认 4 项关键点全面解决
- Point 1 (Security): 详细说明安全性章节新增内容、引用文献与修改位置
- Point 2 (QARMAv2 Baseline): 解释首个位切片实现背景与查找表基准合理性
- Point 3 (Editorial): 列举全部术语修正、图表标记、参考文献与术语一致性决策
- Point 4 (Figures/Tables): 说明 Table 1 格式优化与全部图表矢量化技术细节

**方法论透明度强化** 回复 OPO 算法伪代码/参数查询, 说明 Split()与 SelectBetterPair()函数在 Algorithm 1 已提供充分重实现细节, 补充完整实现开源于 GitHub 仓库([https://github.com/jiahaoxiang2000/bitsliced\\_optimize](https://github.com/jiahaoxiang2000/bitsliced_optimize))包含 LCB 框架、OPO 算法与 BGC 编码, 确保完全可复现性。