# 周报-向嘉豪 (2024-11-18)

向嘉豪

衡阳师范学院

2024 年 11 月 18 日

# 摘要

## 本周工作

本周主要对 OPO（Optimization of Permutation Operation）算法进行了优化和重构：

- 重构 PPO 类结构提高代码可维护性
- 应用于 AES 的 ShiftRow 操作并进行性能测试
- 相比 [SS16] 的实现，性能提升 9.7%

# OPO 算法优化

**Algorithm 1** Optimization of Permutation Operation (OPO)

**Input:** $P$ is a set composed of pairs $(p, m)$, where $p$ is the number of shifts, and $m$ is the mask.
**Output:** $P'$ is an optimized set composed of pairs $(p, m)$.

$P' \leftarrow \emptyset$
**while** $0 \neq \text{len}(P)$ **do**
  $(p, m) \leftarrow \text{MinShift}(P)$ {Find the minimum number of shifts}
  $P \leftarrow P - (p, m)$
  **if** $(p, \cdot)$ in $P'$ **then**
    $(p, m_1) \leftarrow (p, \cdot)$ in $P'$
    $P' \leftarrow P' - (p, m_1)$
    $P' \leftarrow P' + (p, m_1 \vee m)$ {Proposition 1}
  **else**
    **if** Split$((p, m), P')$ is not empty **then**
      $(p_1, m_1), (p_2, m) \leftarrow \text{Split}((p, m), P')$ {Proposition 2}
      $P \leftarrow P + (p_1, m_1)$
      $P \leftarrow P + (p_2, m)$
    **else**
      $P' \leftarrow P' + (p, m)$
    **end if**
  **end if**
**end while**
**return** $P'$

图 1: 原始 OPO 算法

**Algorithm 1** Optimization of Permutation Operations (OPO)

**Input:** Set of pairs $P(p, m)$, optimization index $n$
**Output:** Optimized set of pairs $P'$

$P' \leftarrow P$
**if** $n = \text{Length}(P')$ **then**
  **return** $P'$
**end if**
$P'_1 \leftarrow \text{OPO}(P', n + 1)$
**if** $\text{Split}(P'[n]) \neq \emptyset$ **then**
  $(p_1, m_1), (p_2, m_2) \leftarrow \text{Split}(P'[n])$
  $P' \leftarrow P' \cup \{(p_1, m_1), (p_2, m_2)\}$
  Delete $P'[n]$
  $P'_2 \leftarrow \text{OPO}(P', n)$
**end if**
**return** $\text{Better}(P'_1, P'_2)$

图 2: 优化后 OPO 算法

# PPO 类重构

## 重构目标

- 解决手动计算中间寄存器问题
- 简化汇编转化过程
- 提高代码可维护性

```
class PPO:
    def __init__(self, temp, origin, dest, mask, shift):…

    def __eq__(self, value: "PPO") -> bool:…

    def toList(self):…

    @staticmethod
    def merge(ppo_list):…
```

图 3: PPO 类结构

# ShiftRow 优化对比

## 原始实现

操作序列:

- 7 个 PPO 操作
- 较大的代码体积

## 优化后实现

操作序列:

- 5 个 PPO 操作
- 更紧凑的代码

# 性能对比

表 1: AES 算法实现性能对比

| 实现方案 | 优化等级 | 周期数 | Flash 大小（字节） |
|---|---|---|---|
| [SS16] | O3 | 8,932 | 27,100 |
| 本文工作 | O3 | **8,068** | **25,948** |

## 优化效果

- 执行周期数减少 9.7%
- Flash 内存占用减少

# 参考文献 I


Peter Schwabe and Ko Stoffelen.
All the AES you need on cortex-m3 and M4.
In Roberto Avanzi and Howard M. Heys, editors, *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*, volume 10532 of *Lecture Notes in Computer Science*, pages 180–194. Springer, 2016.

# 老师评语

**标题：Linear Layer Bitsliced Implementation Form & Optimization Schemes & Linear layer**

改为 Bitsliced Implementation of Linear Layers & Proposed Optimization Schemes & Linear Layer Optimization

**全文类似的不突出不明确不清晰的很多，是不是水平就是这样，提高不了了?**

对表述不明确的地方进行修改。

**本周计划**

完成论文初稿，对论文精修。