周报

2025 年 1 月 21 日

摘要

- **实现进展**：完成 SPHINCS$^+$ 算法签名实现
- **分析工作**：研究签名过程的时间复杂度，确定优化方向

FORS 实现

- 将 $8 \times n$ 比特长度的 hm 分割成 k 份
- 构建 k 棵子树，每棵高度为 t
- 计算每棵子树的根节点和认证路径

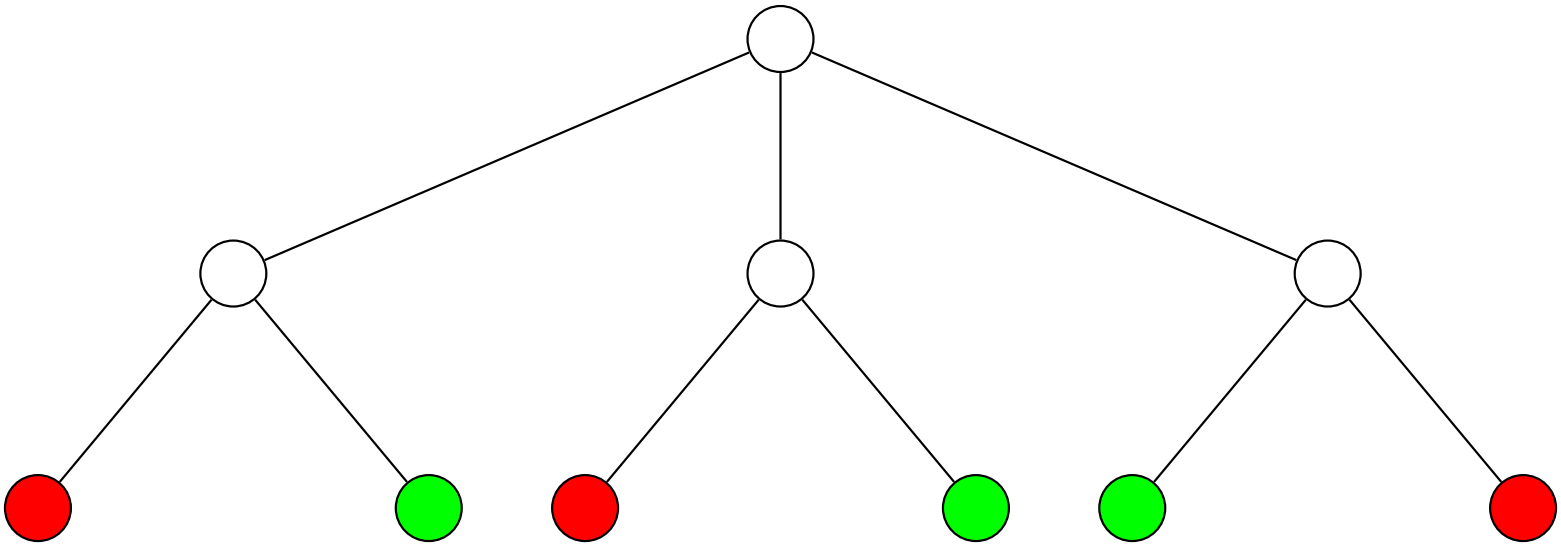

图 1: FORS 树示例 ($k = 3, t = 1$)

HT 树实现

- d 层 Merkle 树结构
- 每层使用 WOTS+ 签名计算
- 最终生成公钥根节点

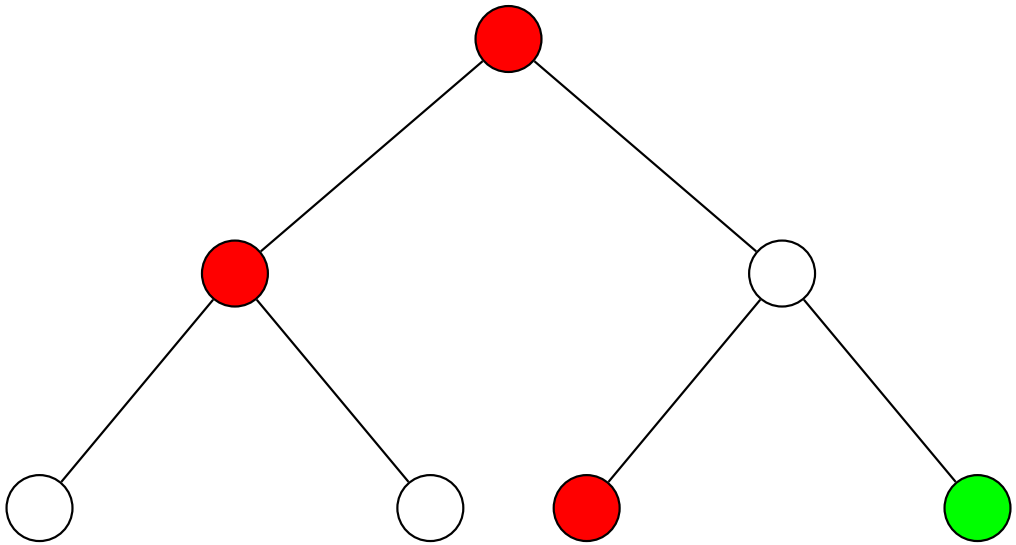

图 2: HT 树示例 ($h' = 1, d = 2$)

时间复杂度分析

FORS 阶段：

- 叶子节点生成：$2^t k$ 次哈希
- 根节点计算：$2^t k + 1$ 次哈希
- 总计约 $16n + 1$ 次哈希运算

HT 阶段：

- WOTS+ 签名：$nw/\log w$ 次哈希/链
- 每层 XMSS 树：$2^{h'}(nw/\log w + 1)$ 次哈希
- 总计约 $d \times 2^{h'}(\frac{nw}{\log w} + 1)$ 次哈希

优化方向

- **并行化重点**：HT 树计算（占主要计算量）
- **具体策略**：
  - FORS 的 k 棵子树并行计算
  - HT 树 d 层 XMSS 树的并行计算
- **预期效果**：在 SPX-128f 配置（$k = 33, d = 22$）下显著提升性能

老师评语

你是对比哪个档次期刊来做这个工作的，参考对比决定了你将来发论文的档次

IEEE transaction of Compute 2024 《CUSPX: Efficient GPU Implementations of Post-Quantum Signature SPHINCS+》

下周计划

- GPU 并行化实现 FORS 和 HT 树
- 推进论文写作