

# 周报

向嘉豪

衡阳师范学院

2025 年 1 月 14 日

- 论文阅读：梳理清楚 SPHINCS<sup>+</sup> 的所有组件
- 实验：Python 实现 WOTS<sup>+</sup> 签名

# SPHINCS<sup>+</sup> (SPX)

SPX 的签名依托 Merkle Hash 树，将多个安全私钥 ( $sk_i$ ) 视为树的叶子节点，多次哈希后得到根节点 ( $PK$ ) 并对外公开，如图 1所示。对于签名方而言，所有  $sk_i$  均可使用，但为了保护私钥安全，只会在签名中暴露必须的中间哈希节点与局部私钥。验证方只需据此重构根节点，与公开的  $PK$  对比一致，即能完成验证。

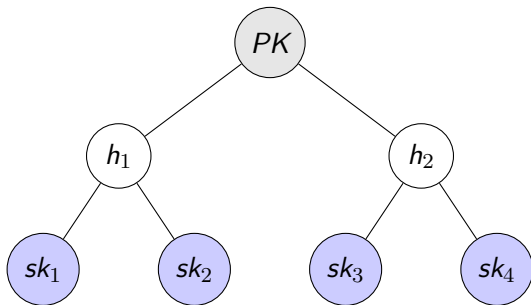


图 1: SPX 中的 Merkle Hash 树结构示意图

# FORS 树

FORS (Forest Of Random Subsets) 树由  $k$  个并排的 Merkle 子树组合而成 (图2)。每个子树根节点用于拼接形成 FORS 的签名  $SIG_{FORS}$ 。在验证环节, 需要公开相应私钥部分以及各子树的中间哈希节点, 以重建并校验每个子树的根节点。FORS 基于消息  $m$  的哈希值, 快速定位并公开对应的  $sk_i$ , 然后将所有子树的根节点拼接成一个整体, 用于后续 HT 树的输入。

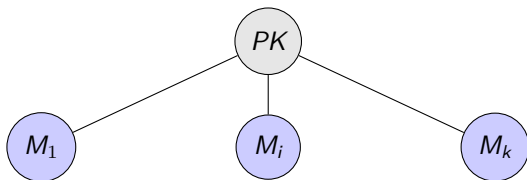


图 2: FORS 树示意图:  $k$  个 Merkle 子树并排组合

# HT 树

HT (Hypertree) 结构采用分层聚合的方式 (图3): 每层 XMSS 树的根节点作为下一层的叶子节点, 最终在顶部生成全局的  $PK$ . 在验证环节, SPX 结合各层子树的 WOTS+ 签名与中间哈希节点来完成验证流程。

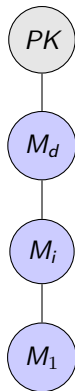


图 3: HT 树示意图: 逐层汇聚得到全局公钥

# Python 实现 WOTS+ 签名

实现进度：主要包括 FORS 签名和 HT 签名，FORS 签名依赖于 Merkle 树的构建和哈希函数的实现，HT 签名以 XMSS 树为基础，同时需要 WOTS+ 签名 XMSS 树的叶子节点。目前已完成 WOTS+ 签名的 Python 实现，如图4所示。

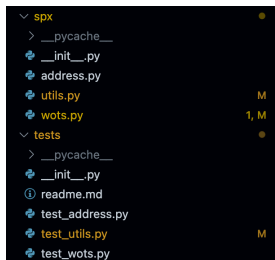


图 4: SPHINCS<sup>+</sup> Python 实现进度

## 继续推进

## 下周计划

- SPHINCS<sup>+</sup> 的完整实现，包括 FORS、HT 树等关键组件
- 进一步研究 SPHINCS<sup>+</sup> 的并行计算特性，探索 GPU 加速优化方案