

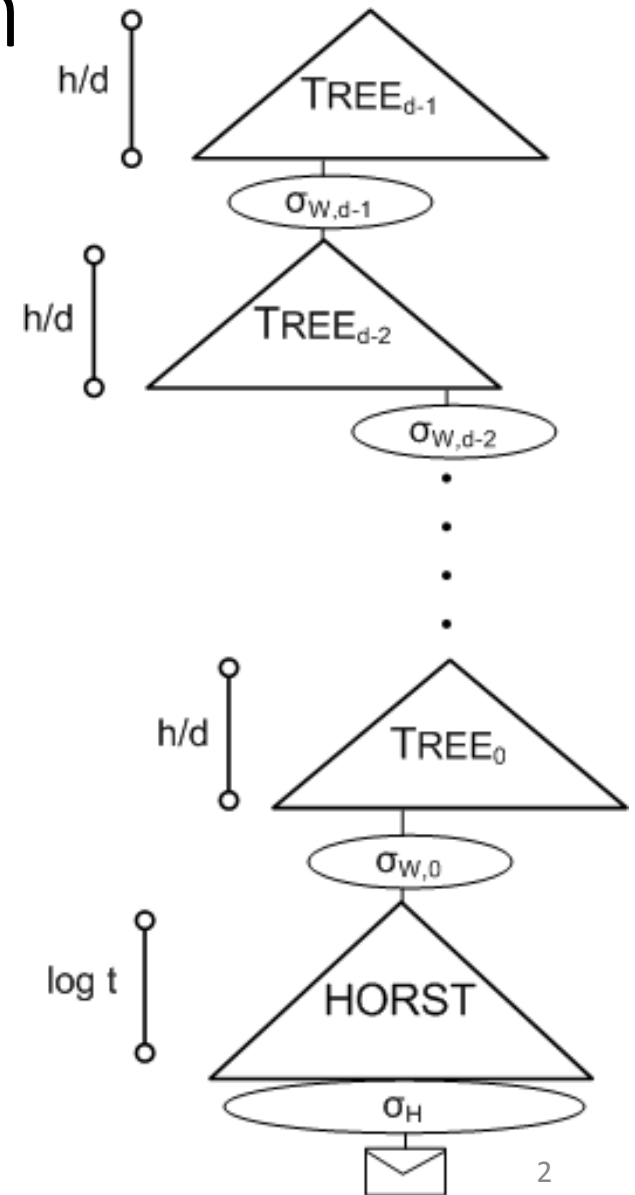
SPHINCS⁺

Submission to the NIST post-quantum project

Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder,
Scott Fluhrer, Stefan-Lukas Gazdag, **Andreas Hülsing**, Panos
Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen,
Florian Mendel, Ruben Niederhagen, Christian Rechberger,
Joost Rijneveld, Peter Schwabe

The SPHINCS Approach

- Use a “hyper-tree” of total height h
- Parameter $d \geq 1$, such that $d \mid h$
- Each (Merkle) tree has height h/d
- (h/d) -ary certification tree

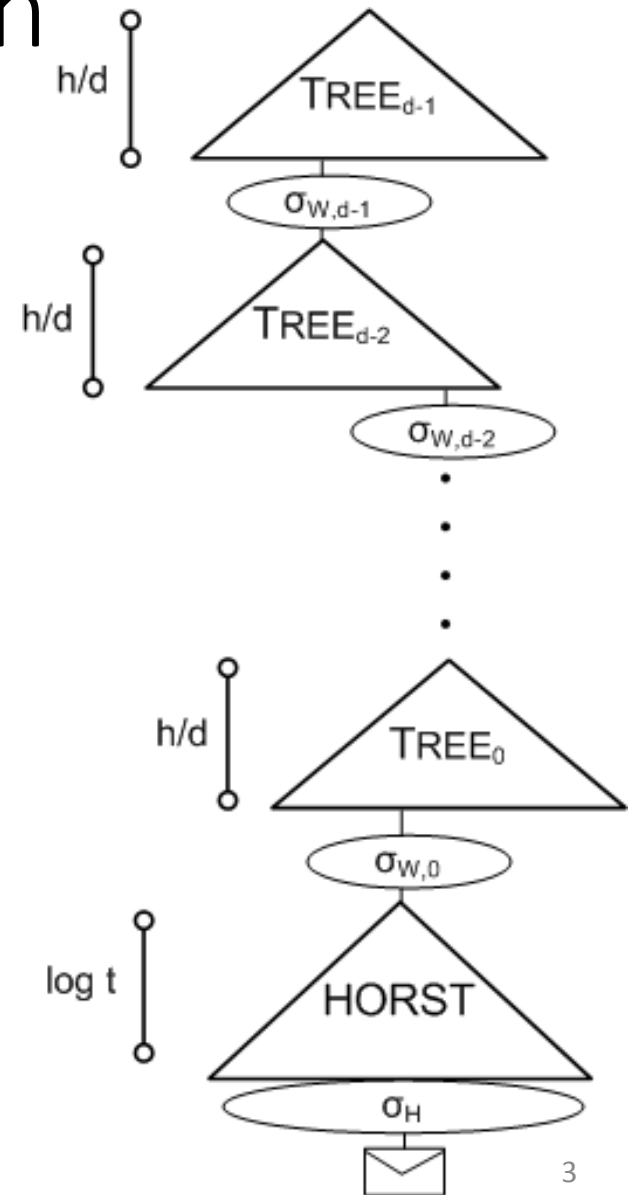


The SPHINCS Approach

- Pick index (pseudo-)randomly
- Messages signed with few-time signature scheme
- Significantly reduce total tree height

- Require

$$\sum_{r \in [0, \infty]} (\Pr[r \text{ -- times index collision}] * \text{Succ}_{\text{EU-CMA}}^{\text{HORST}}(A, q = r)) = \text{negl}(n)$$



SPHINCS⁺ modifications

Adding multi-target attack resilience

- Preimage search:

$$\text{Succ}_{\mathcal{H}_n}^{\text{OW}}(\mathcal{A}) = \left(\frac{q+1}{2^n} \right),$$

- Multi-target preimage search:

$$\text{Succ}_{\mathcal{H}_{n,p}}^{\text{SM-OW}}(\mathcal{A}) = \left(\frac{(q+1)p}{2^n} \right),$$

- Multi-function multi-target preimage search

$$\text{Succ}_{\mathcal{H}_{n,p}}^{\text{MM-OW}}(\mathcal{A}) = \left(\frac{q+1}{2^n} \right),$$

Tweakable hash functions

$$T_l: \mathbb{B}^n \times \mathbb{B}^{32} \times \mathbb{B}^n \rightarrow \mathbb{B}^n,$$
$$\text{md} \leftarrow T_l(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M)$$

- Generates new keys and bitmasks for each call from **PK.seed** and **ADRS**.
- Allows to embed one challenge per call in reduction

Why not collision resistance?

- Bernstein, SHARCS'09:
pq-collision finding costs at least $2^{n/2}$
- Same as cost for pq-(second-)preimage finding?
- **No!** Comparing apples and oranges. Compares cost for pq-(second-)preimage finding in query complexity model to cost for pq-collision finding in more realistic model.
- Also stronger complexity-theoretic assumption!
(Minicrypt vs (conj.) Cryptomania)

FORS

Shortcomings of HORST

- „index collisions“
 - Allows to search for weak messages (no impact on SPHINCS as hash randomized)
 - Still reduces security
- Indices are in unordered list
- Authentication paths will most likely contain redundant nodes
 - Variable size signatures could go lower but requires complicated algorithm (and protocols have to reserve worst-case size) -> see Gravity-SPHINCS's Octopus

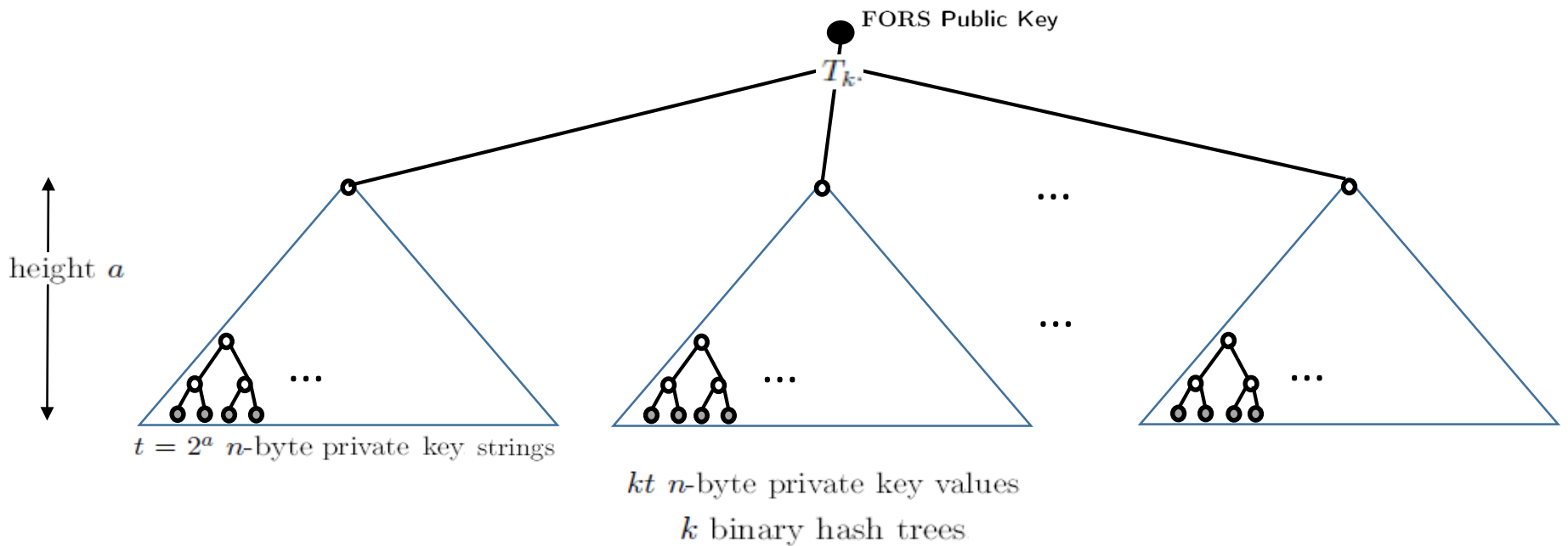
FORS

FORS (Forest of random subsets)

- No index collisions
 - „One tree per index“
- Ordered list of indices
- Signature size same as worst-case variable signature size (at same security level)
 - Only need authpaths in small trees
 - Simple to compute

FORS

- Parameters t , $a = \log t$, k such that $ka = m$



Verifiable index selection

(and optionally non-deterministic randomness)

- SPHINCS:

$$\begin{aligned}(\text{idx}||\mathbf{R}) &= PRF(\mathbf{SK}.prf, M) \\ \text{md} &= H_{\text{msg}}(\mathbf{R}, PK, M)\end{aligned}$$

- SPHINCS⁺:

$$\begin{aligned}\mathbf{R} &= PRF(\mathbf{SK}.prf, \text{OptRand}, M) \\ (\text{md}||\text{idx}) &= H_{\text{msg}}(\mathbf{R}, PK, M)\end{aligned}$$

Optionally non-deterministic randomness

- Non-deterministic randomness complicates side-channel attacks
- Bad randomness in worst-case still leads to secure pseudorandom value

Verifiable index selection

Improves FORS security

- SPHINCS: Attacks could target „weakest“ HORST key pair
- SPHINCS⁺: Every hash query ALSO selects FORS key pair
 - Leads to notion of interleaved target subset resilience

Instantiations

- SPHINCS⁺-SHAKE256
- SPHINCS⁺-SHA-256
- SPHINCS⁺-Haraka

Instantiations (small vs fast)

	n	h	d	$\log(t)$	k	w	bitsec	sec level	sig bytes
SPHINCS ⁺ -128s	16	64	8	15	10	16	133	1	8 080
SPHINCS ⁺ -128f	16	60	20	9	30	16	128	1	16 976
SPHINCS ⁺ -192s	24	64	8	16	14	16	196	3	17 064
SPHINCS ⁺ -192f	24	66	22	8	33	16	194	3	35 664
SPHINCS ⁺ -256s	32	64	8	14	22	16	255	5	29 792
SPHINCS ⁺ -256f	32	68	17	10	30	16	254	5	49 216

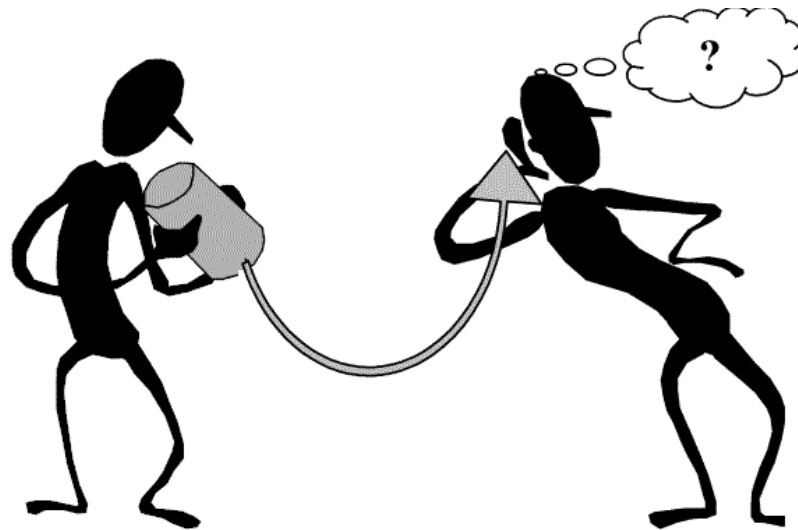
Pro / Con

- Con: Signature size / speed
- Pro: Only secure hash needed
- Pro: Collision-resilient
- Pro: Attacks are well understood (also quantum)
- Pro: Small keys
- Pro: Overlap with XMSS
- Pro: Reuse of established building blocks

Summary of SPHINCS⁺

- Strengthened security gives smaller signatures
- Collision- and multi-target attack resilient
- Fixed length signatures (far easier to compute than Octopus (-> Gravity-SPHINCS))
- Small keys, medium size signatures (lv 3: 17kB)
- Sizes can be much smaller if q_sign gets reduced
- THE conservative choice
- No citable speeds yet

Thank you!
Questions?



Visit us at <https://sphincs.org>