PQCRYPTO
ICT-645622

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

Stateless hash-based signatures

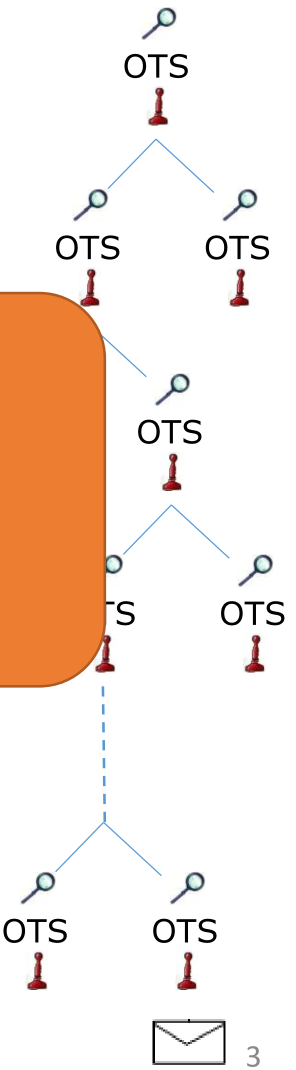
Goldreich

Security parameter $\lambda = 128$

Use binary tree as in Merkle, but...

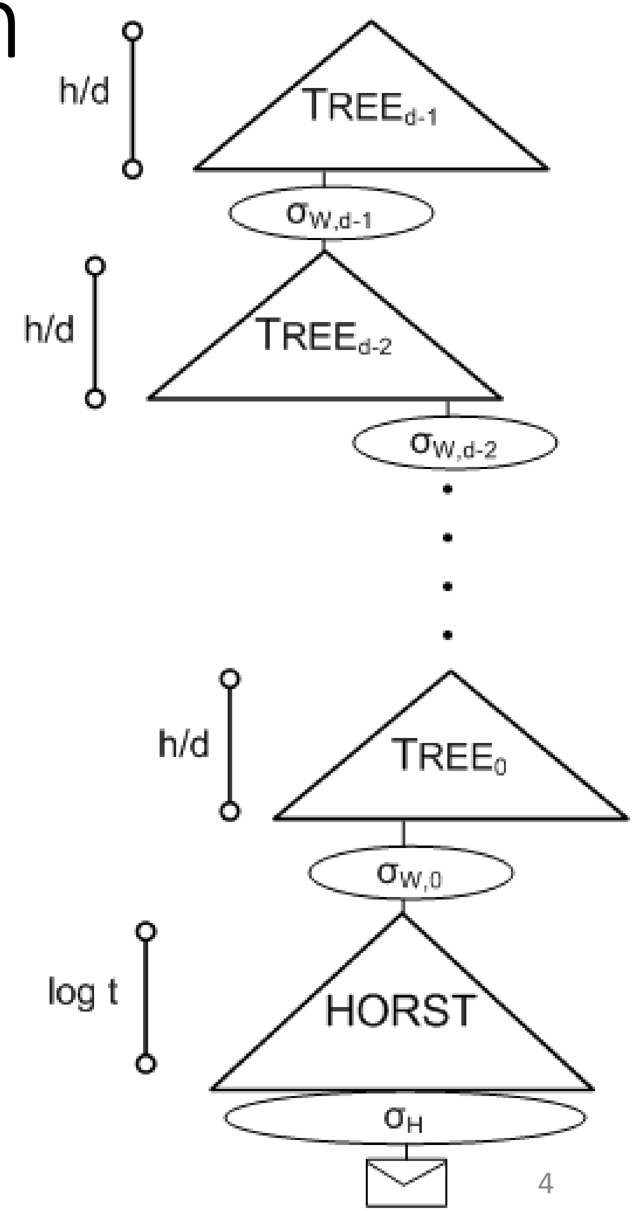
- ...to prevent OTS reuse
  - pick l
  - use h
    collis
- ...for ef
  - use binary certification tree of OTS;
  - all OTS secret keys are generated pseudorandomly.

Even with optimization
(using WOTS-16 as OTS):
0.6 MB signature.



The SPHINCS Approach

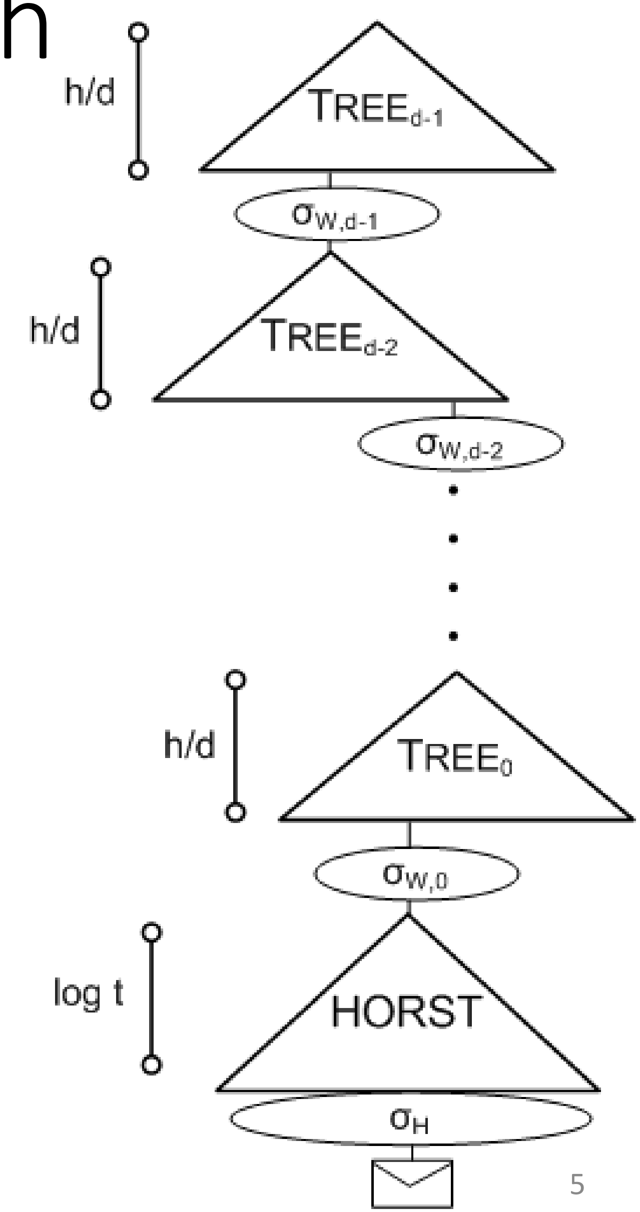
- Use a "hyper-tree" of total height h
- Parameter $d \geq 1$, such that $d \mid h$
- Each (Merkle) tree has height h/d
- (h/d)-ary certification tree



The SPHINCS Approach

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

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



SPHINCS$^+$

Adding multi-target attack resilience

- Preimage search:

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

- Multi-target preimage search:

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

- Multi-function multi-target preimage search

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

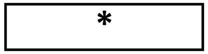
Tweakable hash functions

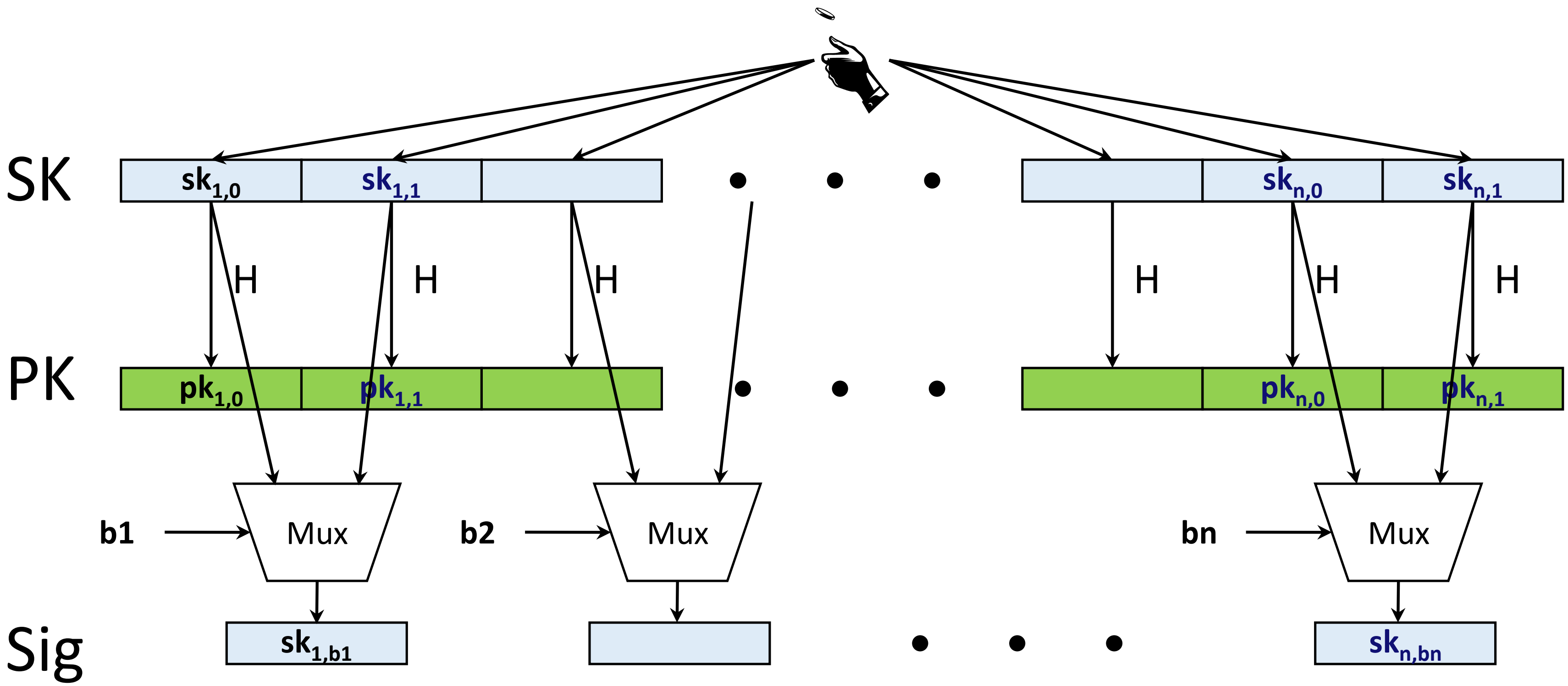
$$T_l\colon\ \mathbb{B}^n \times \mathbb{B}^{32} \times \mathbb{B}^n \rightarrow \mathbb{B}^n,$$
$$\mathrm{md} \leftarrow T_l(\mathbf{PK}.\mathrm{seed}, \mathbf{ADRS}, M)$$

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

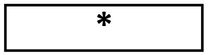
Few-Time Signature Schemes

Recap LD-OTS

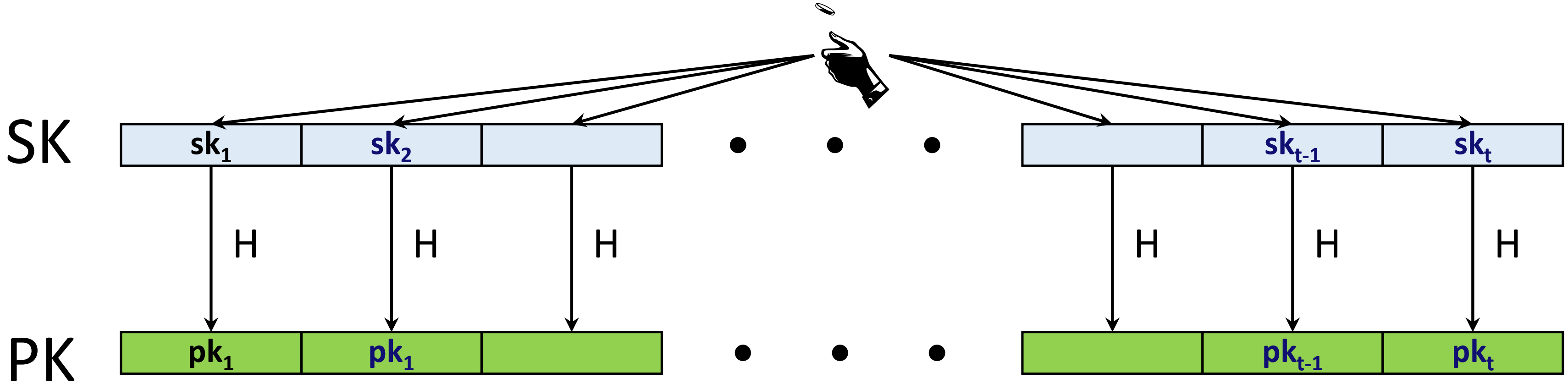
Message M = b1,…,bn, OWF H  = n bit



HORS [RR02]

Message M, OWF H, CRHF H’  = n bit

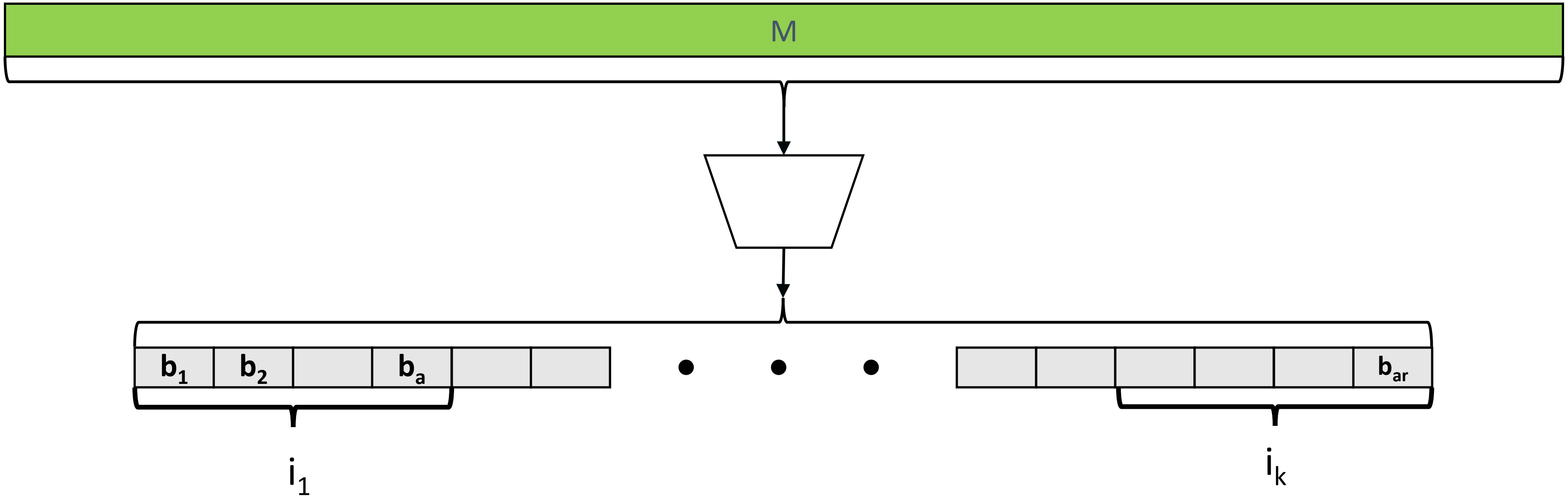
Parameters $t=2^a$,k, with m = ka (typical a=16, k=32)



HORS mapping function

Message M, OWF H, CRHF H' $\boxed{\quad * \quad} = n$ bit

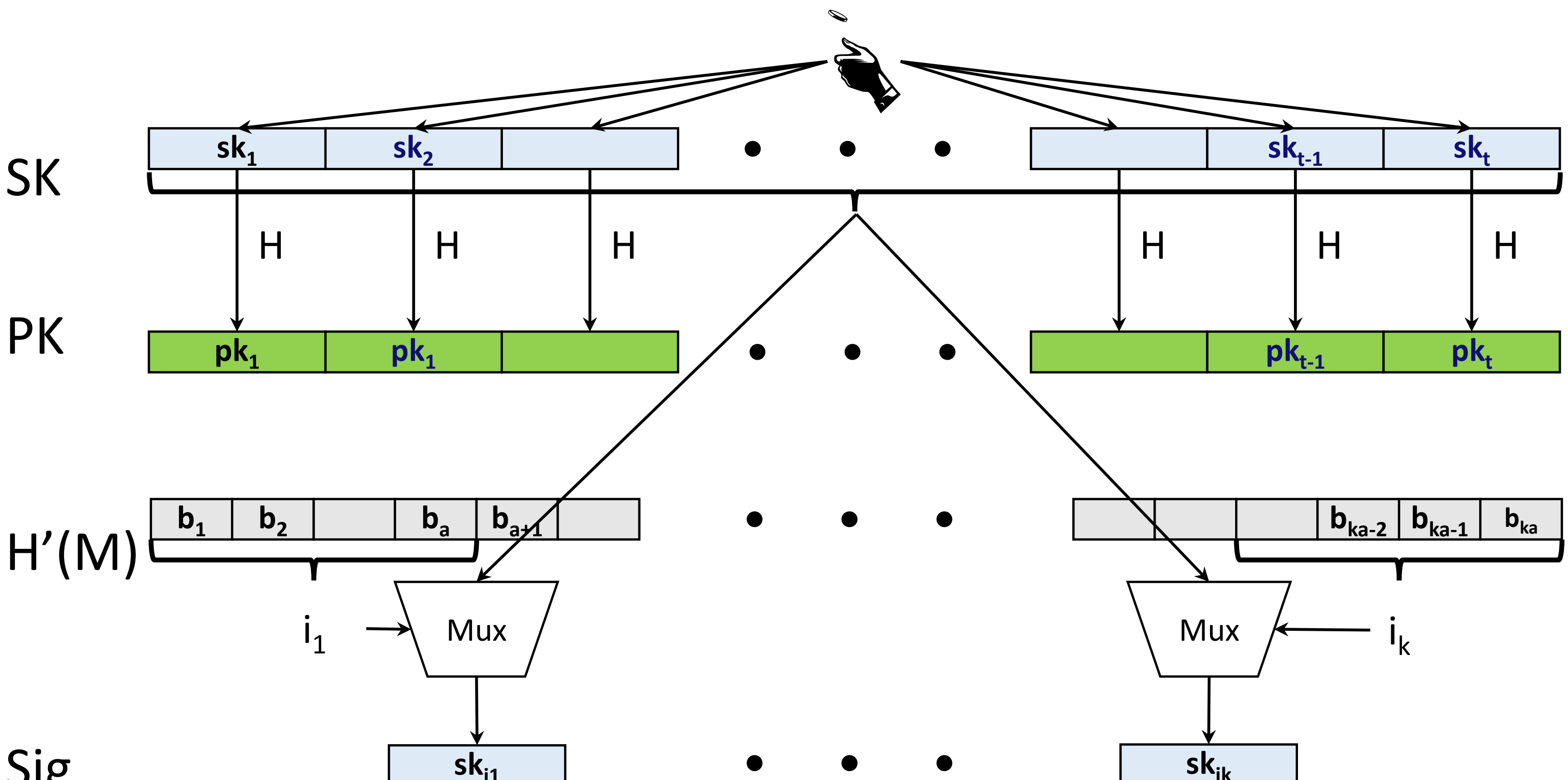
Parameters $t = 2^a, k$, with $m = ka$ (typical $a = 16, k = 32$)



HORS

Message M, OWF H, CRHF H’ [*] $= n$ bit

Parameters $t = 2^a, k$, with $m = ka$ (typical $a = 16, k = 32$)



HORS Security

- M mapped to k element index set $M^i \in \{1, \ldots, t\}^k$
- Each signature publishes k out of t secrets
- Either break one-wayness or…

- r-Subset-Resilience: After seeing index sets M_j^i for r messages $msg_j, 1 \leq j \leq r$, hard to find $msg_{r+1} \neq msg_j$ such that $M_{r+1}^i \in \bigcup_{1 \leq j \leq r} M_j^i$.

- Best generic attack: $\mathrm{Succ}_{\text{r-SSR}}(A,q) = q(rk/t)^k$

→ Security shrinks with each signature!

HORST

Using HORS with MSS requires adding PK (tn bits) to MSS signature. (SPHINCS-256: $n = 256, t = 2^{16}$, $k = 32$)

HORST: Merkle Tree on top of HORS-PK

- New PK = Root
- Publish Auth-Paths for HORS signature values
- PK can be computed from Sig

- With optimizations: $tn \rightarrow (k(\log t - x + 1) + 2^x)n$
  - E.g. SPHINCS-256: 2 MB → 16 KB
- Use randomized message hash

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)

FORS

FORS (Forest of random subsets)

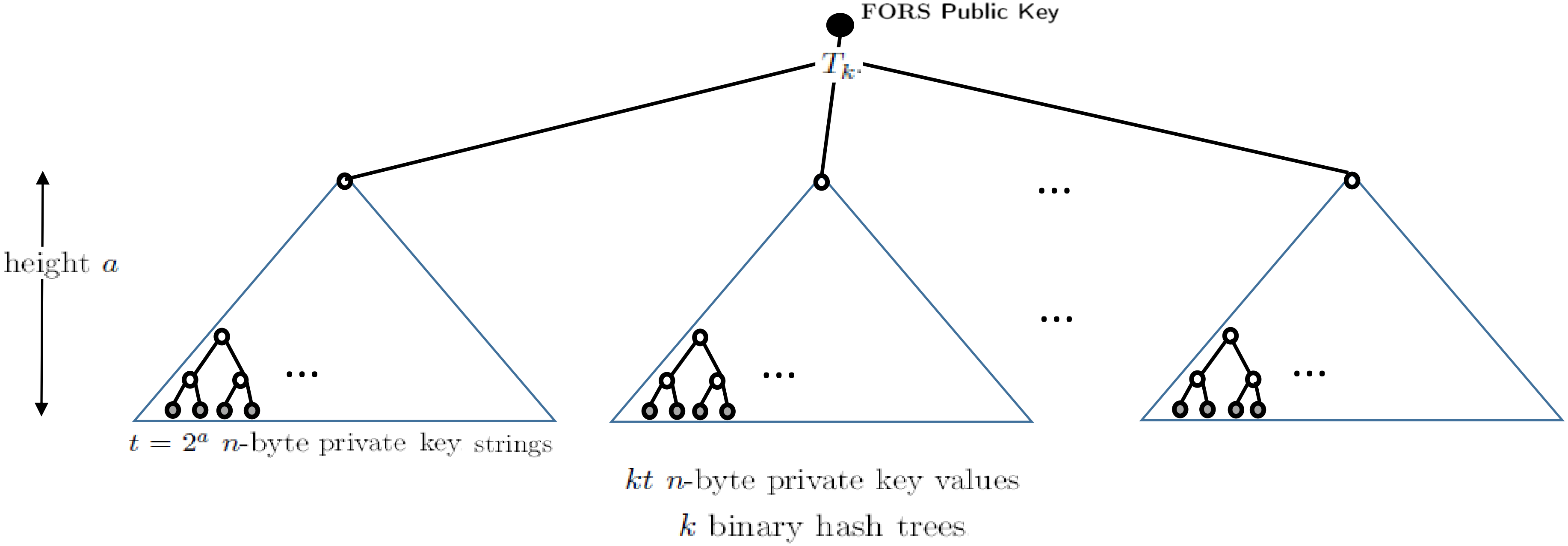
- No index collisions
  - „One tree per index“

- 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:

$$(\mathrm{idx}||\mathbf{R}) = PRF(\mathbf{SK}.\mathrm{prf}, M)$$
$$\mathrm{md} = H_{\mathrm{msg}}(\mathbf{R}, \mathrm{PK}, M)$$

- SPHINCS$^+$:

$$\mathbf{R} = PRF(\mathbf{SK}.\mathrm{prf}, \mathrm{OptRand}, M)$$
$$(\mathrm{md}||\mathrm{idx}) = H_{\mathrm{msg}}(\mathbf{R}, \mathrm{PK}, M)$$

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

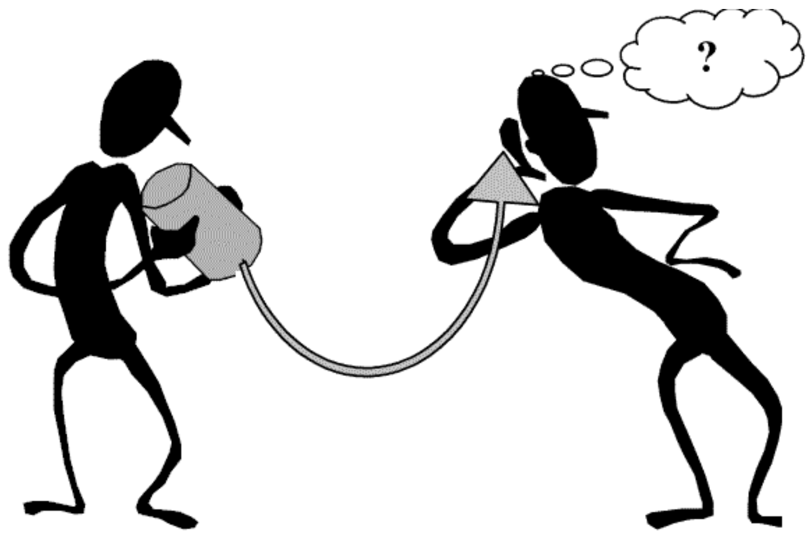
Instantiations (small vs fast)

	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

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