(To be) standardized hash-based signature schemes

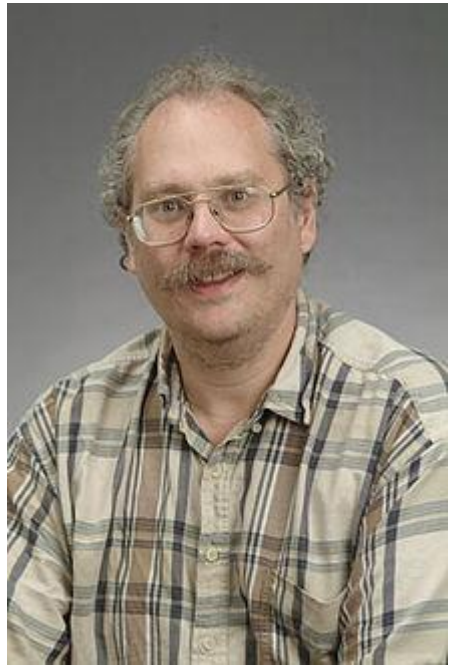
Andreas Hülsing
Eindhoven University of Technology

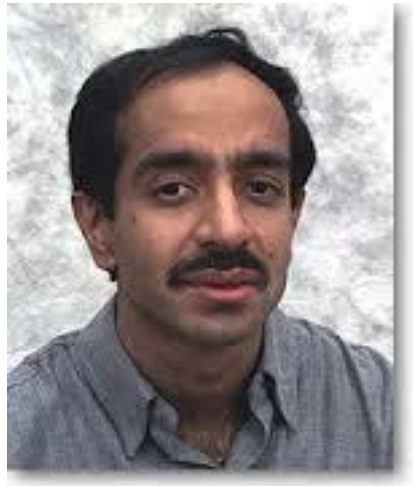
SSR 2018

The quantum threat

- **Shor's algorithm** breaks RSA, (EC)DSA, (EC)DH,…

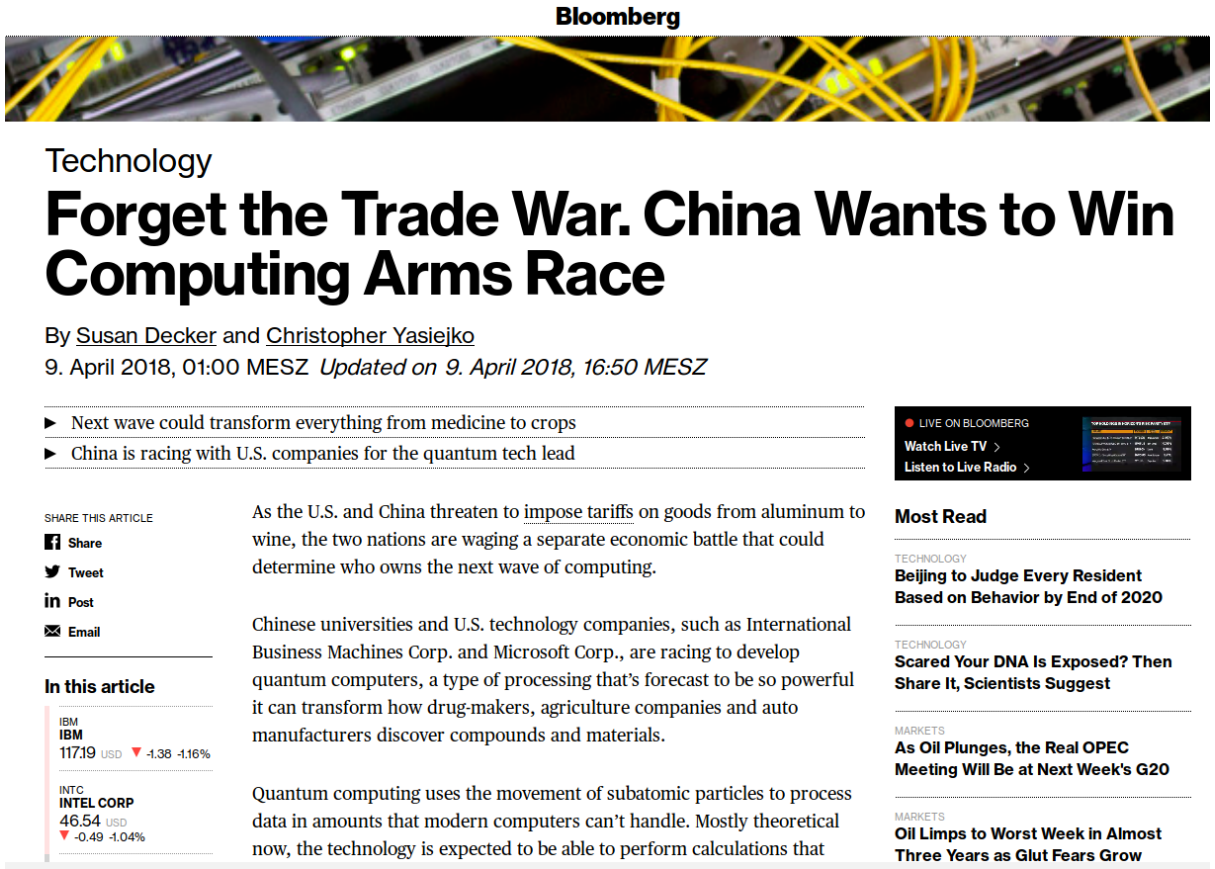
- **Grover's algorithm** asymptotically reduces complexity of brute-force search attacks by a square-root factor.





Why care today

- **EU** launched a one billion Euro project on quantum technologies
- Similar range is spent in **China**
- **US** administration passed a bill on spending $1.275 billion US dollar on quantum computing research
- **Google**, **IBM**, **Microsoft**, **Alibaba**, and others run their own research programs.



Bloomberg

Technology

Forget the Trade War. China Wants to Win Computing Arms Race

By Susan Decker and Christopher Yasiejko

9. April 2018, 01:00 MESZ *Updated on 9. April 2018, 16:50 MESZ*

- Next wave could transform everything from medicine to crops
- China is racing with U.S. companies for the quantum tech lead

SHARE THIS ARTICLE

Share

Tweet

Post

Email

In this article

IBM
IBM
117.19 USD ▼ -1.38 -1.16%

INTC
INTEL CORP
46.54 USD
▼ -0.49 -1.04%

As the U.S. and China threaten to impose tariffs on goods from aluminum to wine, the two nations are waging a separate economic battle that could determine who owns the next wave of computing.

Chinese universities and U.S. technology companies, such as International Business Machines Corp. and Microsoft Corp., are racing to develop quantum computers, a type of processing that's forecast to be so powerful it can transform how drug-makers, agriculture companies and auto manufacturers discover compounds and materials.

Quantum computing uses the movement of subatomic particles to process data in amounts that modern computers can't handle. Mostly theoretical now, the technology is expected to be able to perform calculations that

LIVE ON BLOOMBERG
Watch Live TV
Listen to Live Radio

Most Read

TECHNOLOGY
Beijing to Judge Every Resident Based on Behavior by End of 2020

TECHNOLOGY
Scared Your DNA Is Exposed? Then Share It, Scientists Suggest

MARKETS
As Oil Plunges, the Real OPEC Meeting Will Be at Next Week's G20

MARKETS
Oil Limps to Worst Week in Almost Three Years as Glut Fears Grow

It‘s a question of risk assessment

Long-lived systems

- Development time easily 10+ years
- Lifetime easily 10+ years

- At least make sure you got a secure update channel!

Merkle‘s hash-based signatures
[Lam79,Mer89]

No new hardness assumptions

Provably (post-quantum) secure if (post-quantum) secure hash function is used

Basic concept extremely easy

Stateful

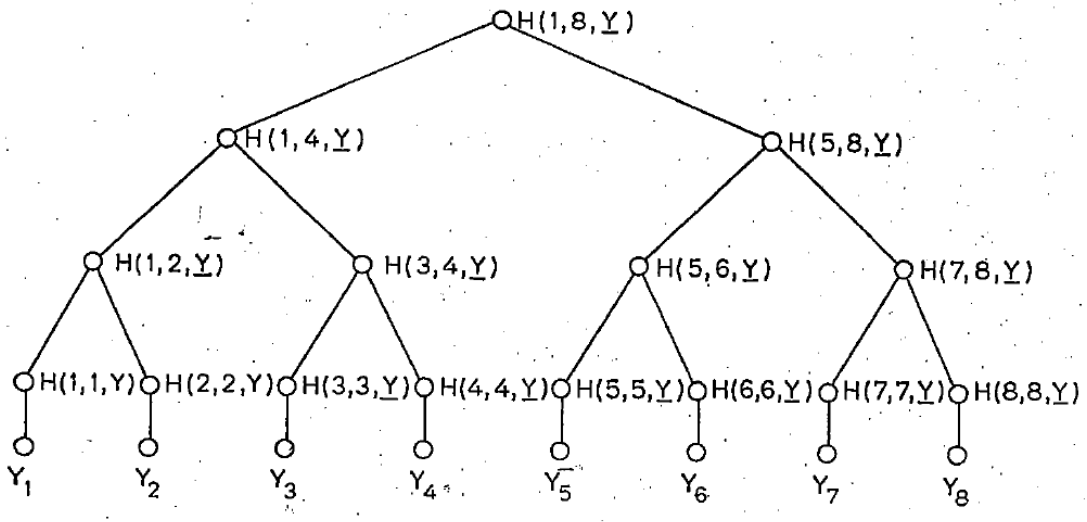
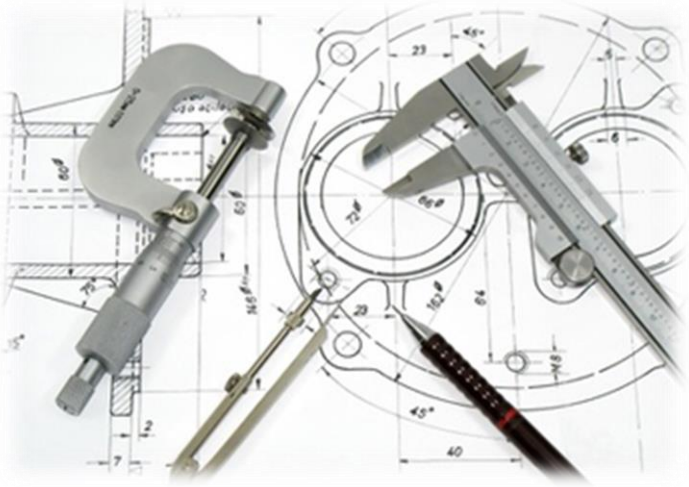


FIG 1
AN AUTHENTICATION TREE WITH N = 8.

PAGE 41B

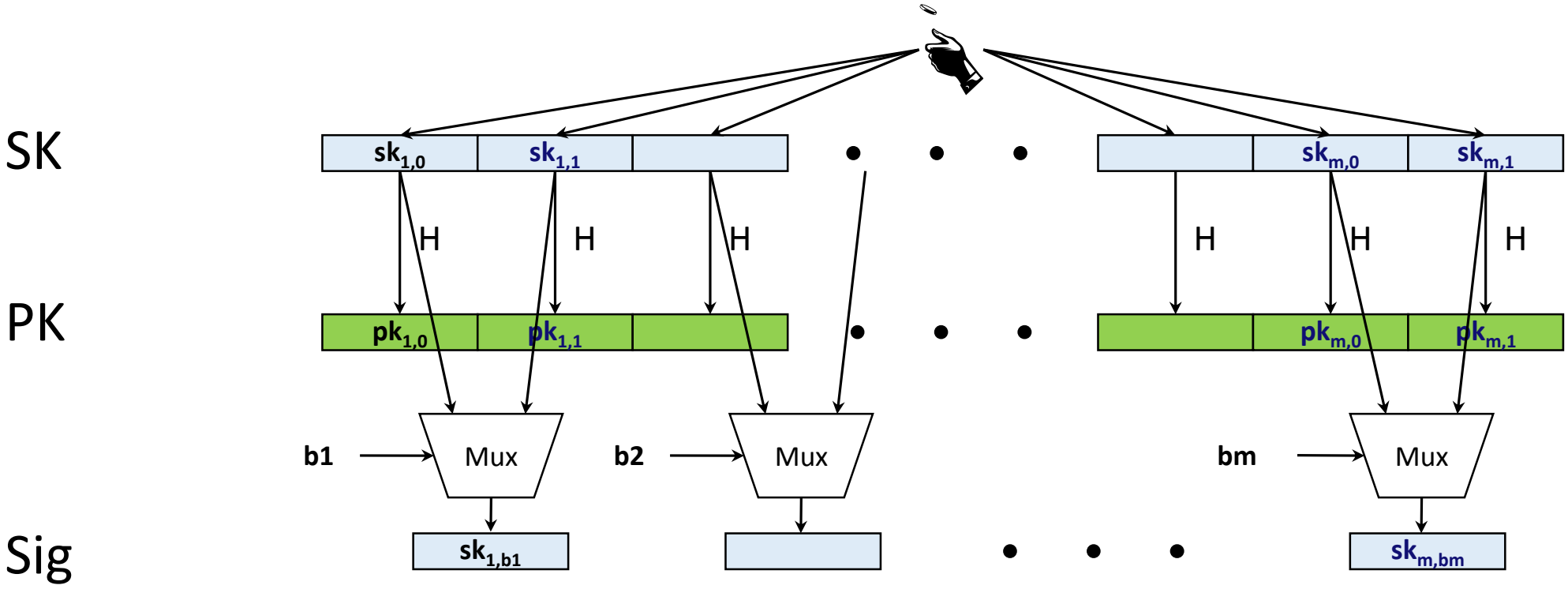
Basic Construction



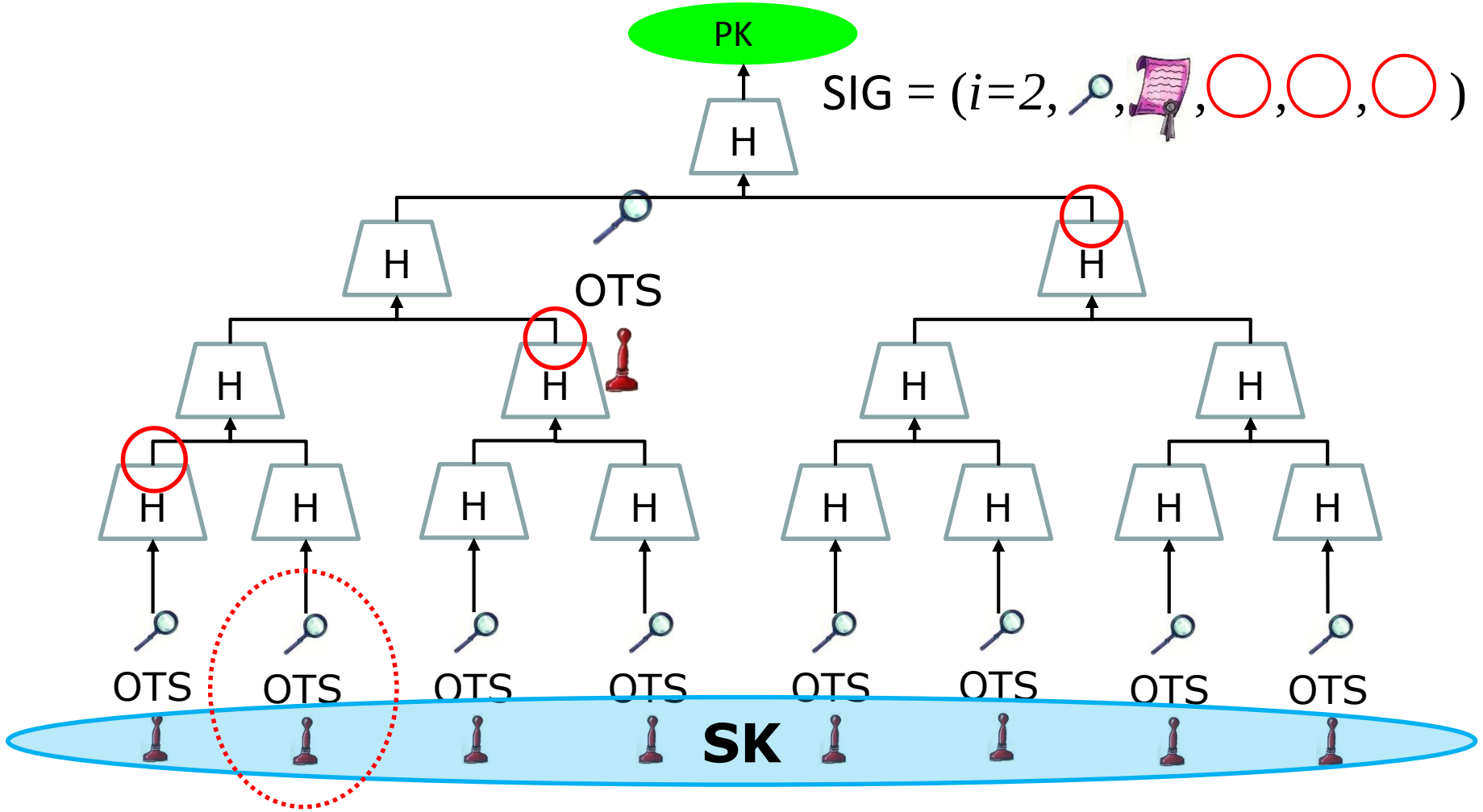
Lamport OTS [Lam79]

Message M = b1,…,bm, OWF H

= n bit



Merkle's Hash-based Signatures



Winternitz-OTS

Lamport-OTS in MSS

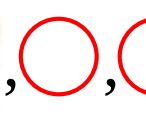
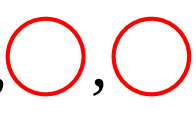
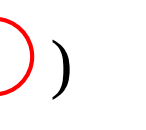
SIG = (i=2,  ,  ,  ,  , )

Verification:

1. Verify 
2. Verify authenticity of 

We can do better!

WOTS in MSS

SIG = (i=2,  ,  ,  , )

Verification:

1. Compute  from 
2. Verify authenticity of 

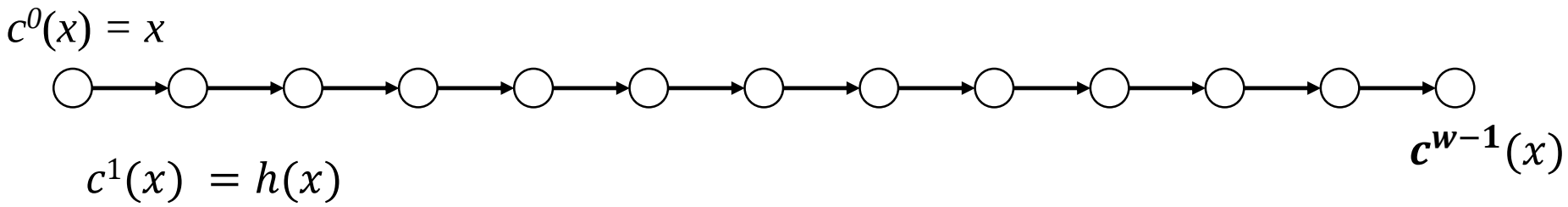
Steps 1 + 2 together verify 

Function chains

Hash function $h : \{0,1\}^n \rightarrow \{0,1\}^n$

Parameter w

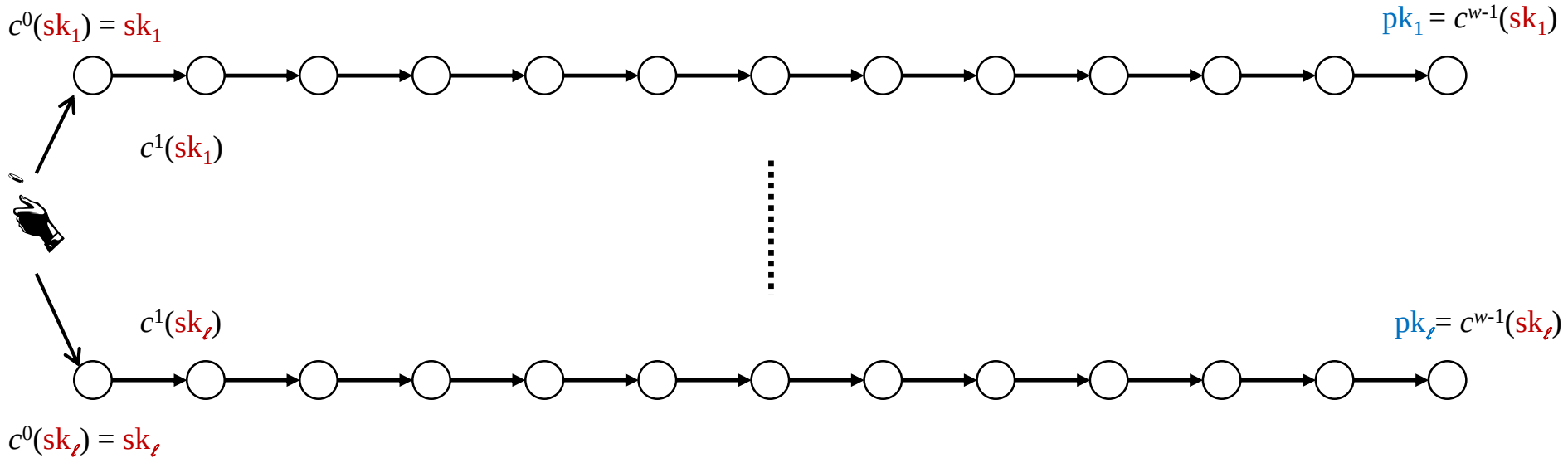
Chain: $c^i(x) = h\left(c^{i-1}(x)\right) = \underbrace{h \circ h \circ \cdots \circ h}_{i\text{-times}}$



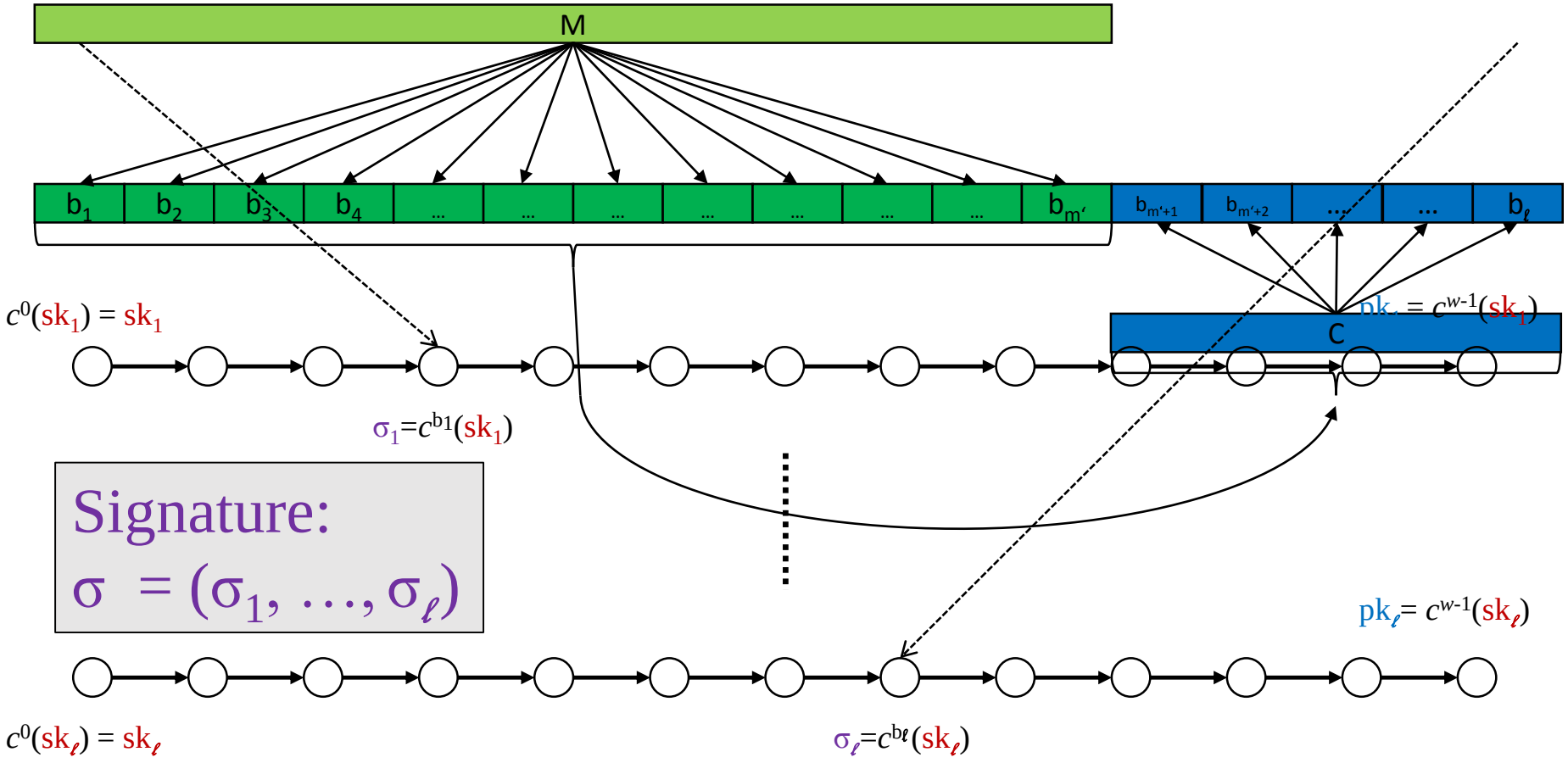
WOTS

Winternitz parameter w (usually a power of 2), security parameter n, message length m, hash function h

Key Generation: Compute l, sample h_k



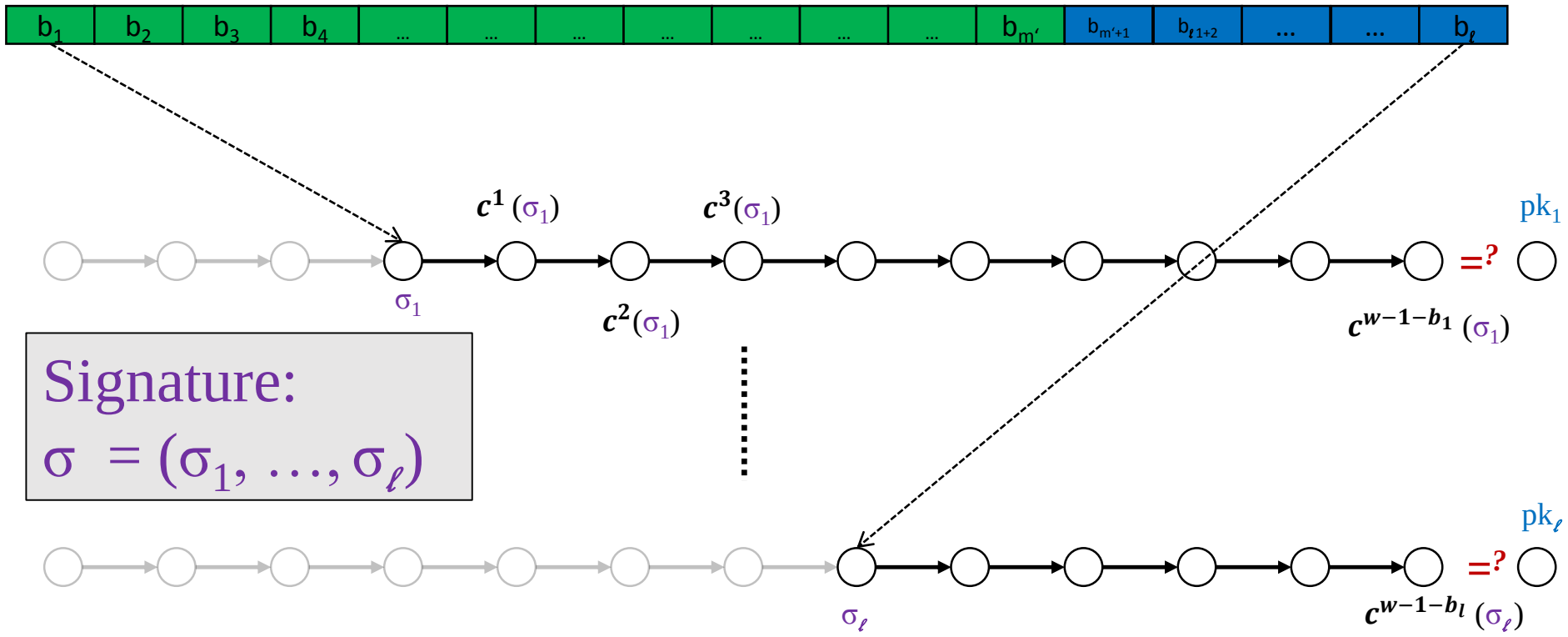
WOTS Signature generation



Signature:
$\sigma = (\sigma_1, \ldots, \sigma_\ell)$

WOTS Signature Verification

Verifier knows: M, w



Multi-Tree MSS

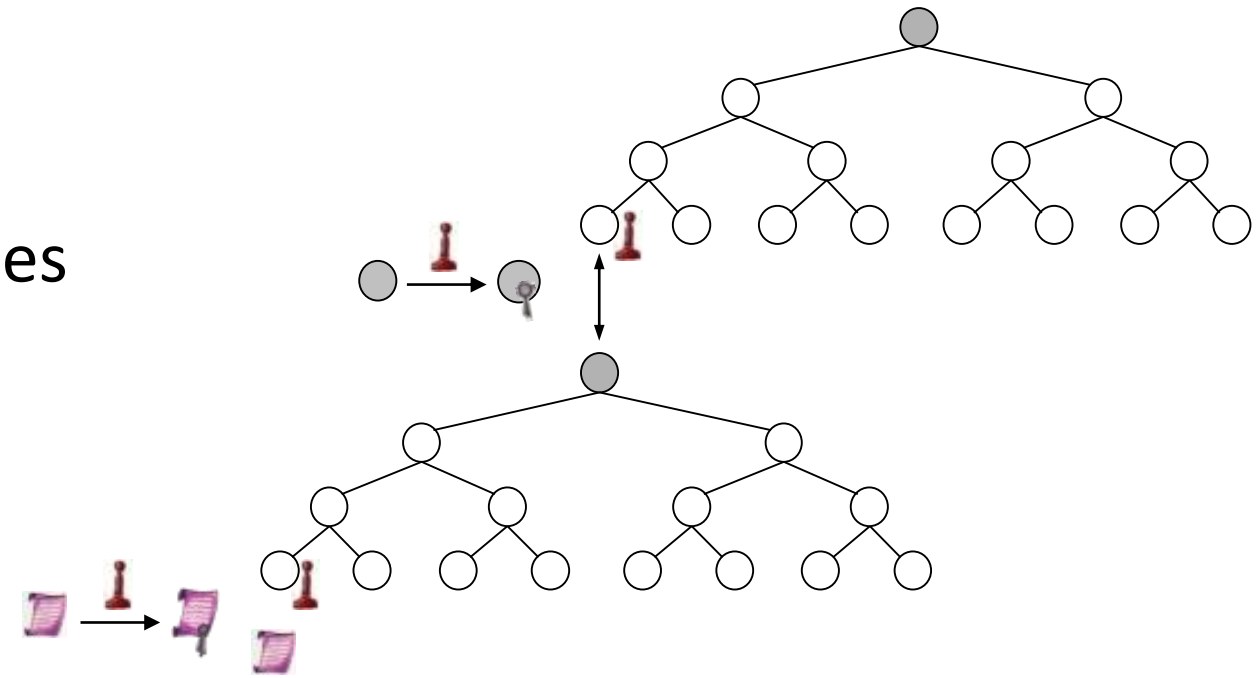
Multi-Tree MSS

Uses multiple layers of trees to reduce key generation time

-> Key generation
(= Building first tree on each layer)
$\Theta(2^h) \rightarrow \Theta(d*2^{h/d})$

-> Worst-case signing times
$\Theta(\mathrm{h}/2) \rightarrow \Theta(h/2d)$

-> Increases signature size by d-1 one-time signatures



Standardization in CFRG (adoption by NIST announced)

Competing proposals in IRTF's CFRG

XMSS (RFC 8391)

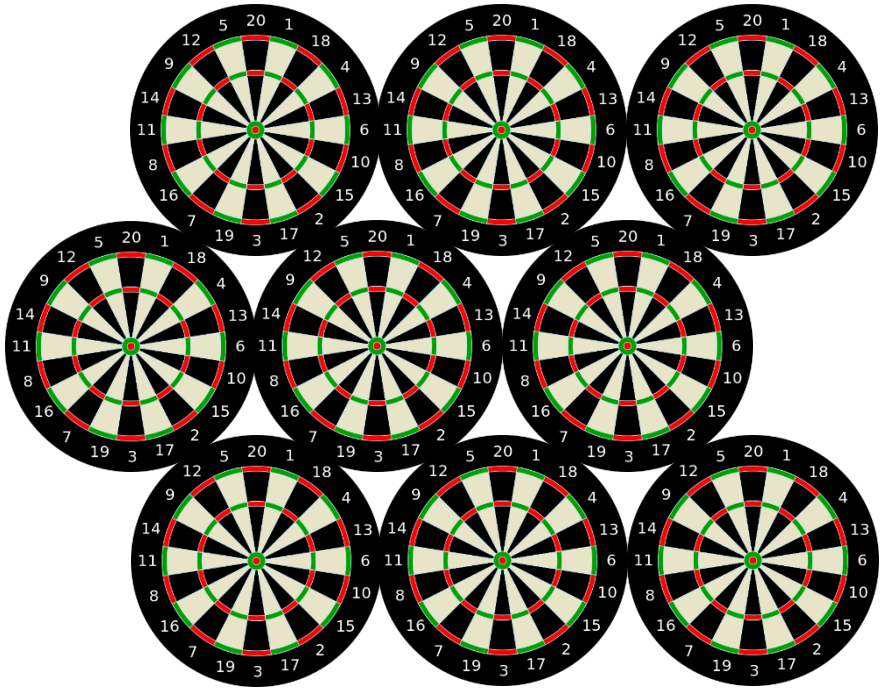
- WOTS + MSS + Multi-tree (WOTS-T + XMSS-T + MT)
- w = 16
- h = 10, 16, 20
- 8 MT (h,d) combinations (trees on all layers have same height)
- SHA2-256 required; SHA2-512, SHAKE-128, SHAKE-256 optional

LMS (draft-mcgrew-hash-sigs-13)

- WOTS + MSS + Multi-tree (WOTS-LM + LMS + HSS=MT)
- w = 2, 4, 16, 256
- h = 5, 10, 15, 20, 25
- 6 HSS combinations with two trees of independent height
- Only SHA2-256

Intermezzo: Multi-target attacks

- WOTS & Lamport need hash function h to be one-way
- Multi-tree of total height 60 with WOTS (w=16) leads $> 2^{60} \cdot 67 \approx 2^{66}$ images.
- Inverting one of them allows existential forgery (at least massively reduces complexity)
- q-query brute-force succeeds with probability $\Theta\left(\frac{q}{2^{n-66}}\right)$ conventional and $\Theta\left(\frac{q^2}{2^{n-66}}\right)$ quantum
- We loose 66 bits of security! (33 bits quantum)

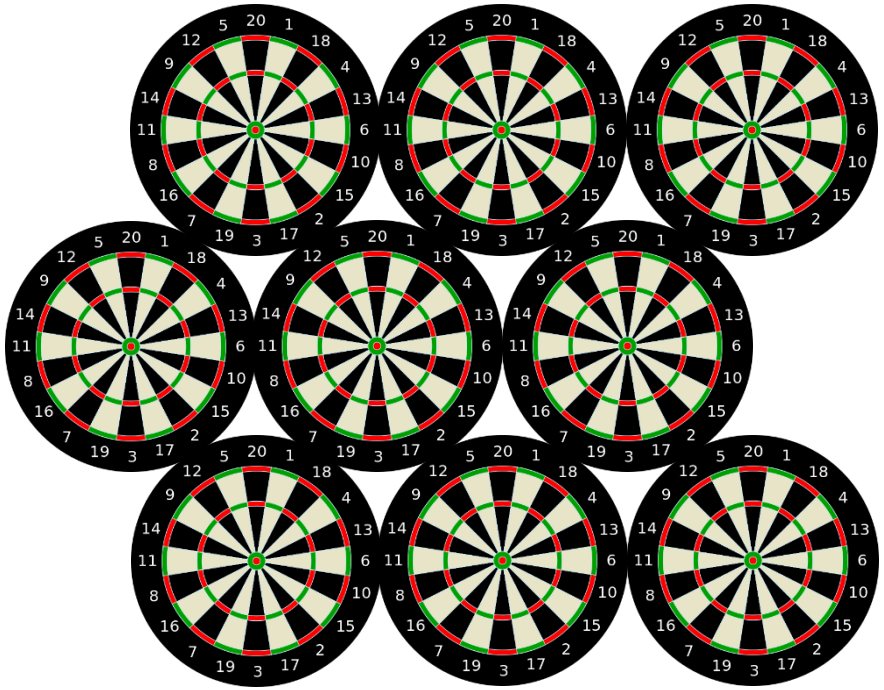


New abstraction: Tweakable Hash Function

- Mitigation: Separate targets
- Common approach:
  - In addition to hash function and „input“ take
    - Hash „Address“ (uniqueness in key pair)
    - Hash „key“ used for all hashes of one key pair (uniqueness among key pairs)
- Tweakable Hash Function:

$$T\ (PP, TW, MSG) \rightarrow MD$$

PP: Public parameters
TW: Tweak
MSG: Message
MD: Message Digest



Security of tweakable hashes

- Necessary notion: Single-function multi-target-collision resistance for distinct tweaks
- Intuition:
  - Adversary gets black box access to $T(PP, \cdot, \cdot)$ for random PP.
  - Adversary can adapatively query with restriction to use each tweak only once.
  - Adversary receives PP and has to find second-preimage for one of its previous queries (such that PP and TW are the same).
- For length-preserving case: A property to guarantees that a preimage finder outputs a „new preimage“. E.g., assume that T is almost regular. (Subject of ongoing research)

Instantiating the tweakable hash (for SHA2)

XMSS

- K = SHA2(pad(PP)||TW),
  BM = SHA2(pad(PP)||TW+1),
  MD= SHA2(pad(K)||MSG $\oplus$ BM)
- Standard model proof if K & BM were random,
- (Q)ROM proof when generating K & BM as above (modeling those SHA2 invocations as RO)
- Tight proof is currently under revision

LMS

- MD = SHA2(PP||TW||MSG)
- QROM proof assuming SHA2 is QRO
- ROM proof assuming SHA2 compression function is RO
- Proofs are essentially tight

Instantiating the tweakable hash

- LMS is factor 3 faster but leads to slightly larger signatures at same security level
- LMS makes somewhat stronger assumptions about the security properties of the used hash function
- More research on direct constructions needed

Stateless hash based signatures
SPHINCS

About the statefulness

- Works great for some settings

- However....

  ... back-up

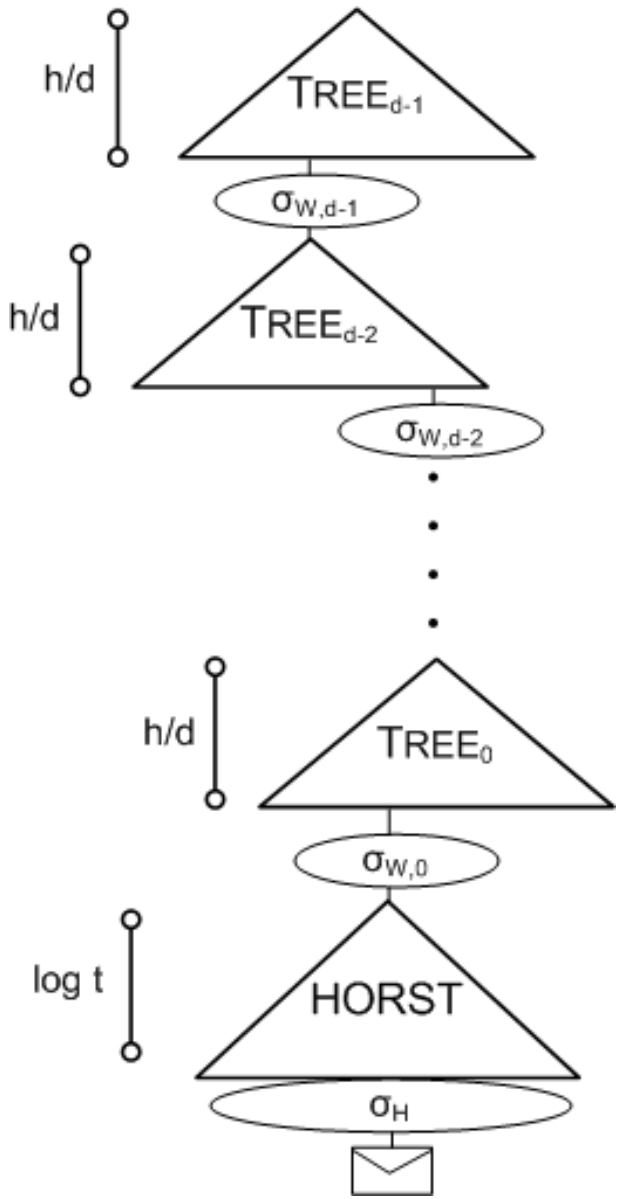
  ... multi-threading

  ... load-balancing

ELIMINATE THE STATE

SPHINCS (2015)

- Stateless Scheme
- XMSS^{MT} + HORST
  + (pseudo-)random index
- Collision-resilient
- Deterministic signing
- SPHINCS-256:
  - 128-bit post-quantum secure
  - Hundrest of signatures / sec
  - 41 kb signature
  - 1 kb keys

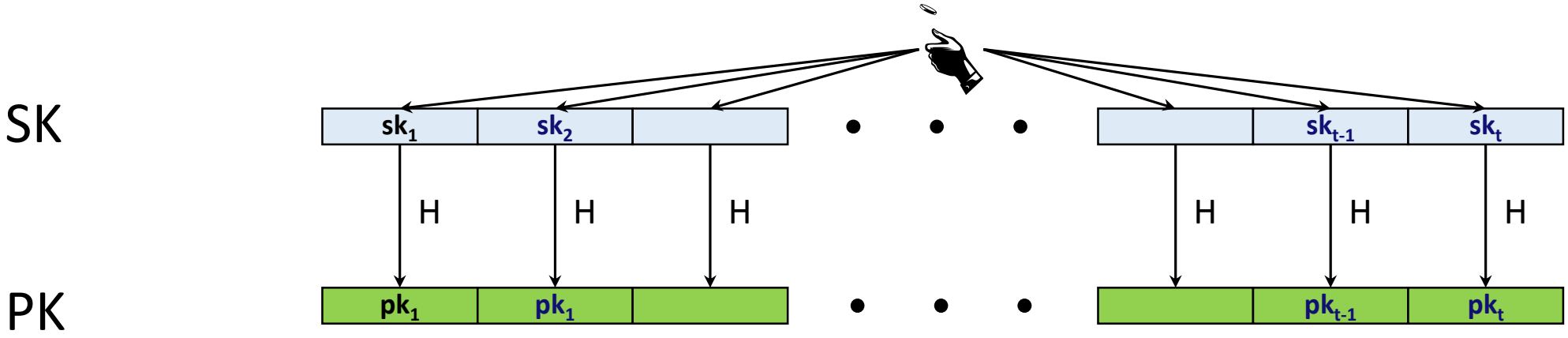


Few-Time Signature Schemes

HORS [RR02]

Message M, OWF H, CRHF H’ [*] = n bit

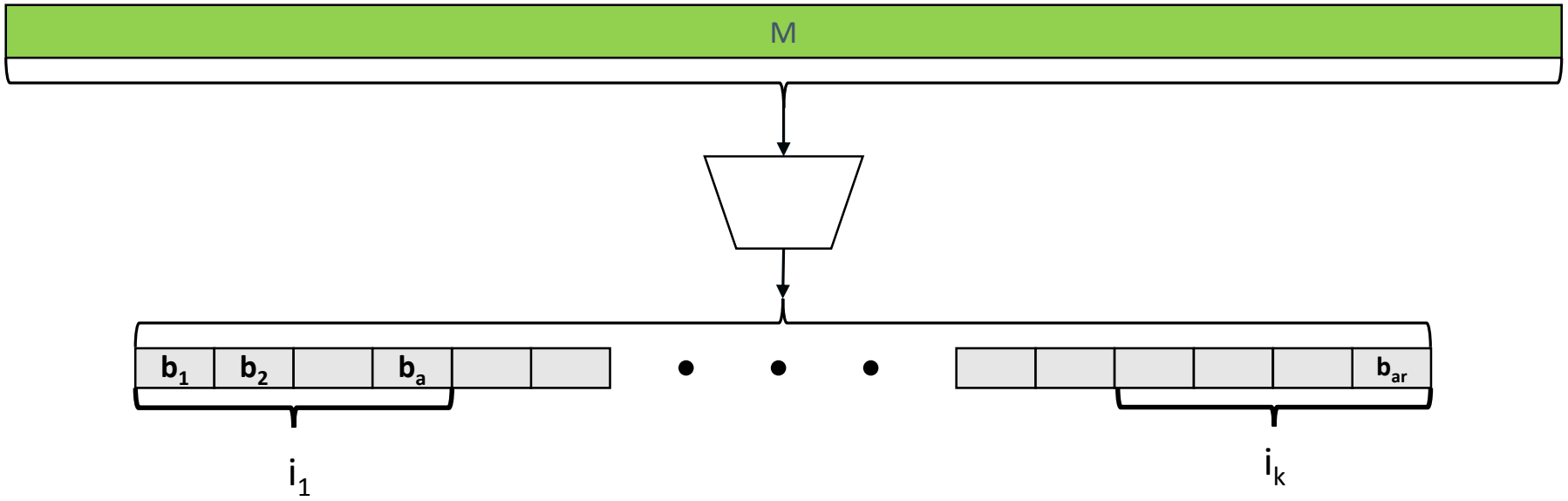
Parameters $t=2^a$,k, with m = ka (typical a=16, k=32)



HORS mapping function

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

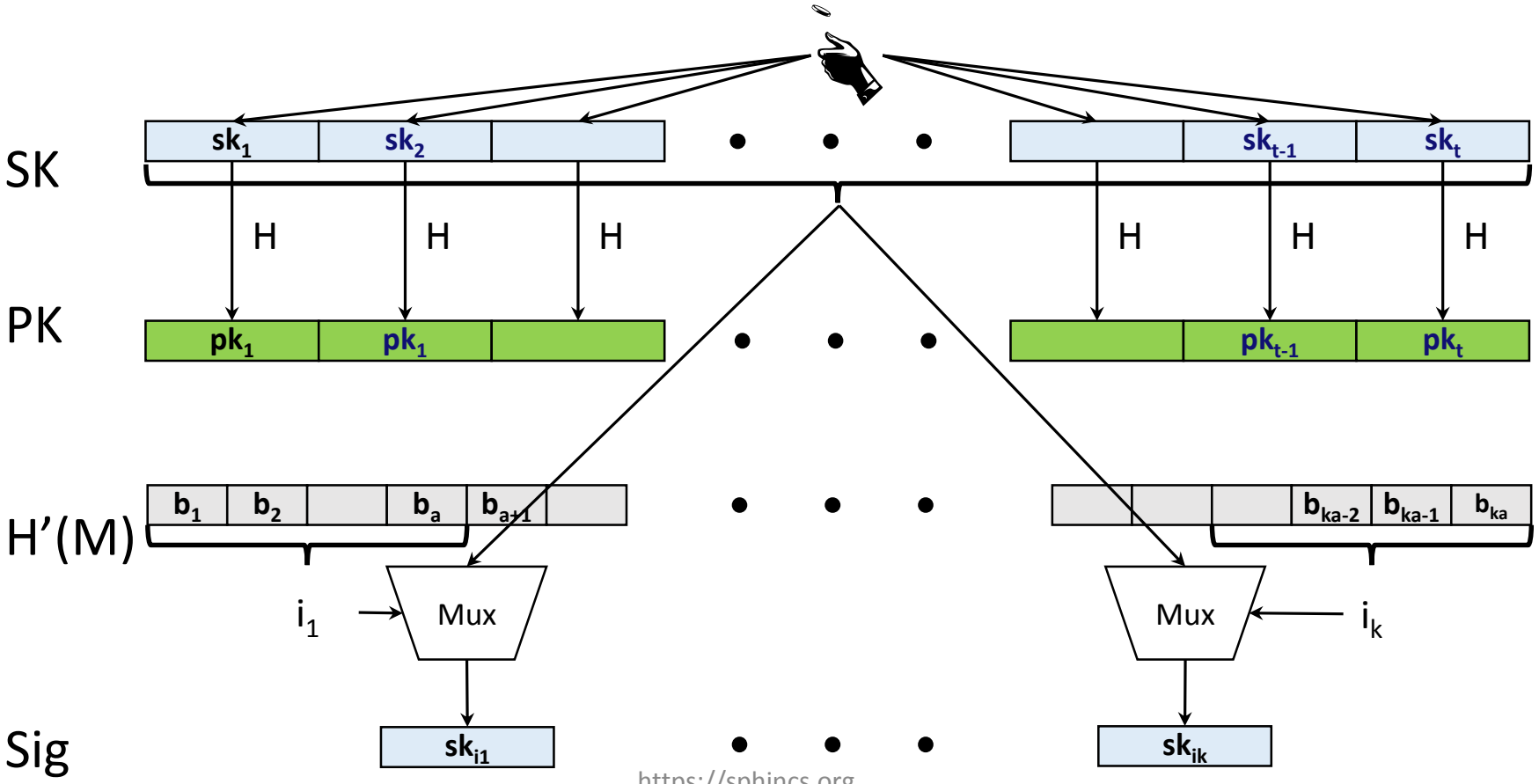
Parameters $t = 2^a, k$, with $m = ka$ (typical $a = 16, k = 32$)



HORS

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

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



HORS Security

- M mapped to k element index set $M^i \in \{1, \dots, t\}^k$
- 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: $\mathrm{Succ}_{\text{r-SSR}}(A,q) = q(rk/t)^k$

$\longrightarrow$ 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 $\rightarrow$ 16 KB
- Use randomized message hash

SPHINCS$^+$

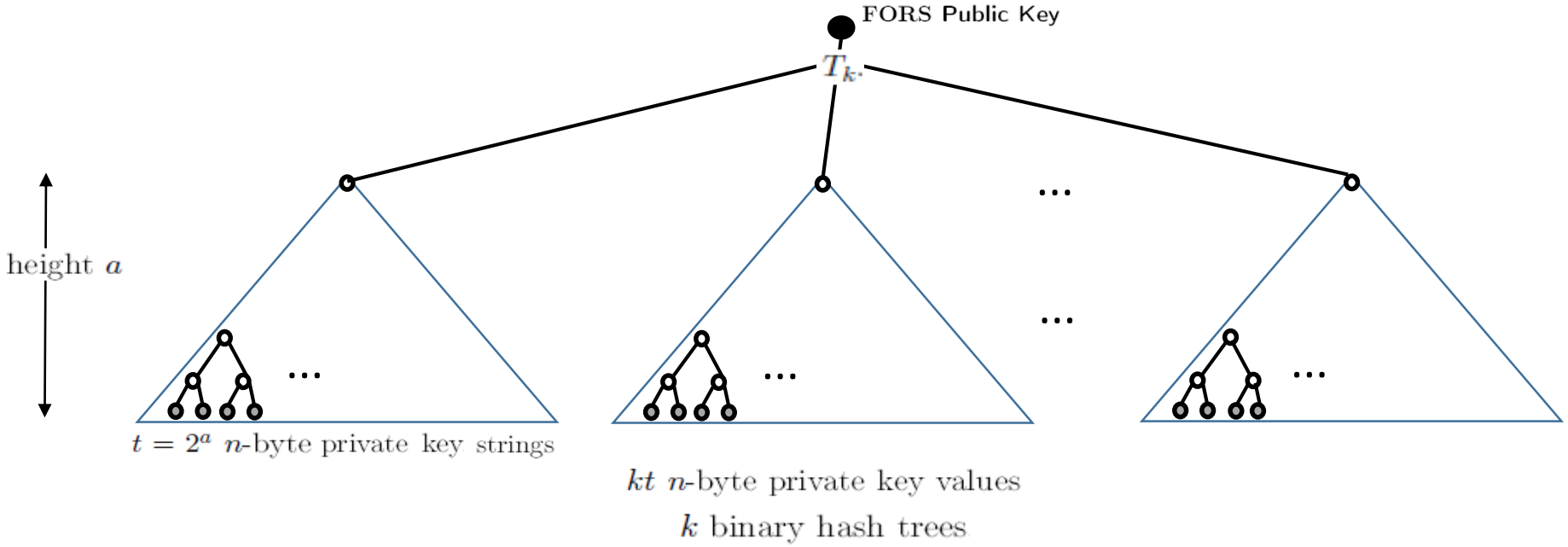
Joint work with Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe

SPHINCS$^+$ (our NIST submission)

- Strengthened security gives smaller signatures
- Collision- and multi-target attack resilient (XMSS tweakable hash)
- 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

FORS (Forest of random subsets)

- 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^{+}:

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

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

Comparison to SPHINCS-128 at same security level

	Signing median cycles	Verifying median cycles	Signature bytes
SPHINCS$^+$ (n = 24, h = 55, d = 11, b = 8, k = 30, w = 16)	67 017 940	1 911 684	21 288
SPHINCS$^+$ (n = 24, h = 51, d = 17, b = 9, k = 30, w = 16)	40 117 282	2 724 094	29 256
SPHINCS-128 (n=32, h=60, d=12, t=2^16, k=32, w = 16)	51 636 372	1 451 004	41 000

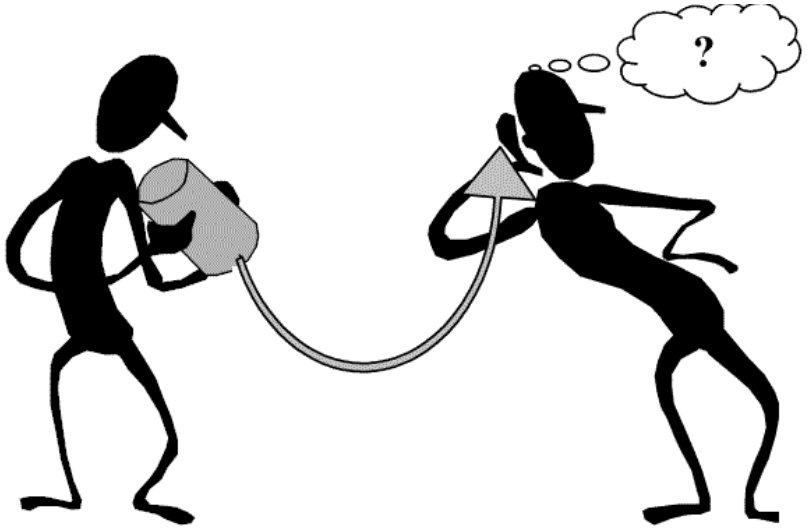
Hash-based Signatures in NIST „Competition“

- SPHINCS$^+$
  - FORS as few-time signature
  - XMSS-T tweakable hash
- Gravity-SPHINCS
  - PORS as few-time signature
  - Requires collision-resistance -> no tweakable hash
- (PICNIC)

Conclusion

- Standardized hash-based signatures are available
  - Both got their pros & cons (independent comparison would be nice)
  - If you can deal with state: Please play around (and tell us about)
- For stateless
  - A lot of trade-offs
  - Need requirements to fine-tune
  - Further comparison is needed (not easy to compare security)
  - Fault-attacks (and mitigations) need further attention
- Tweakable hash functions
  - Might be interesting also in other settings
  - Dedicated constructions would be nice

Thank you!
Questions?



For references, literature & longer lectures see https://huelsing.net