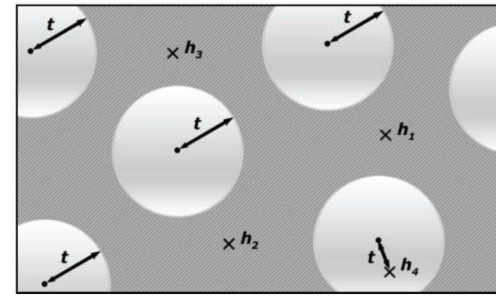
Hash-based Signatures

Andreas Hülsing

CAST Workshop
03/05/2018
Darmstadt

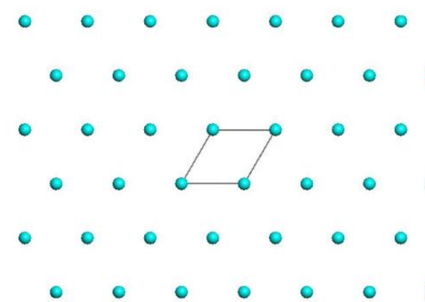
Post-Quantum Signatures

Lattice, MQ, Coding



- Signature and/or key sizes
- Runtimes
- Secure parameters

$$y_1 = x_1^2 + x_1x_2 + x_1x_4 + x_3$$
$$y_2 = x_3^2 + x_2x_3 + x_2x_4 + x_1 + 1$$
$$y_3 = \ldots$$



Hash-based Signature Schemes

[Mer89]

- Post quantum
- Only secure hash function
- Security well understood
- Fast

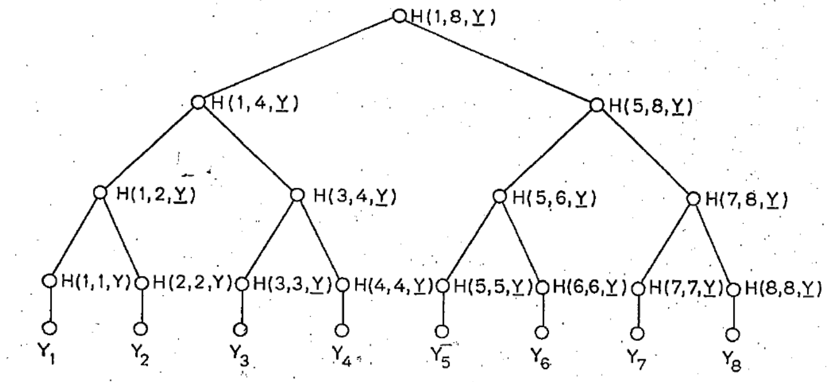
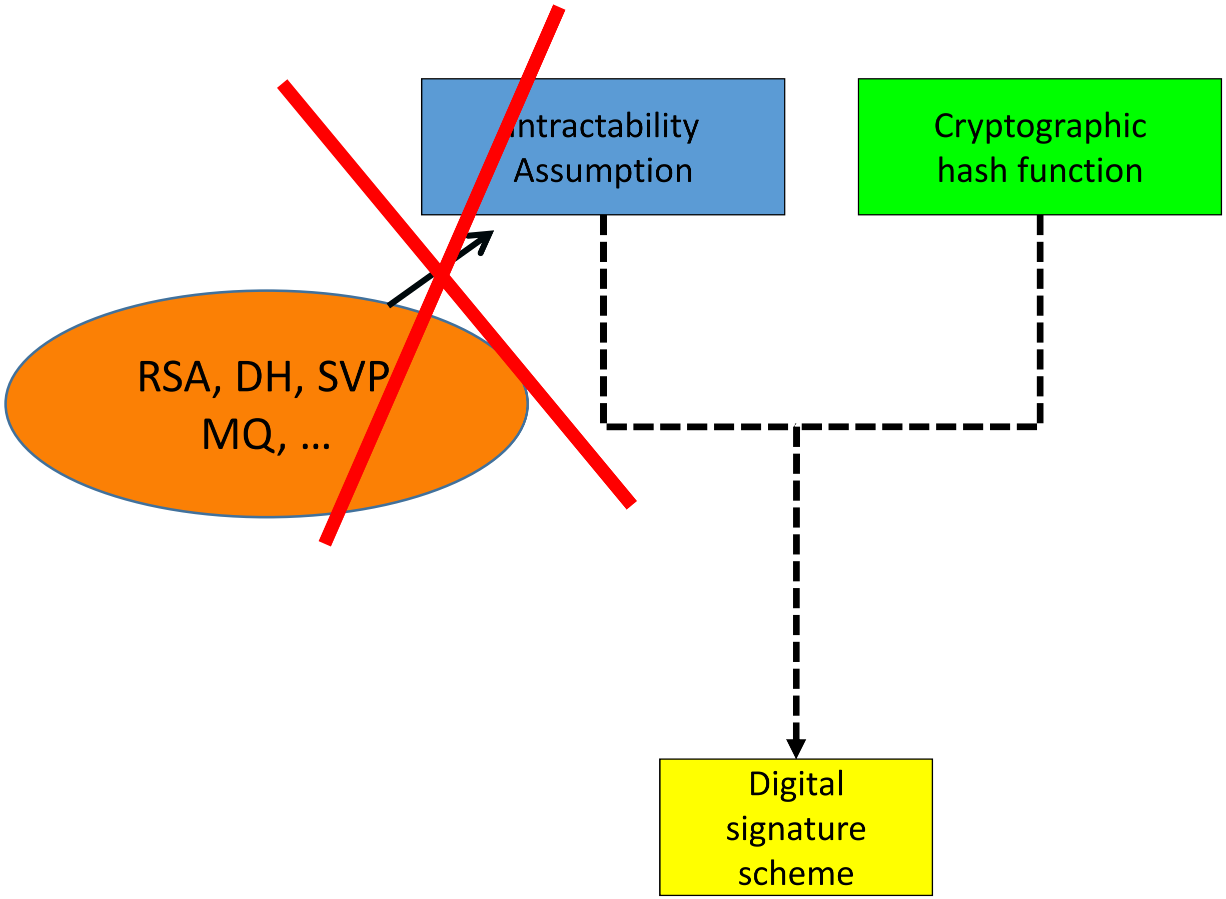


FIG 1

AN AUTHENTICATION TREE WITH N = 8.

PAGE 41B

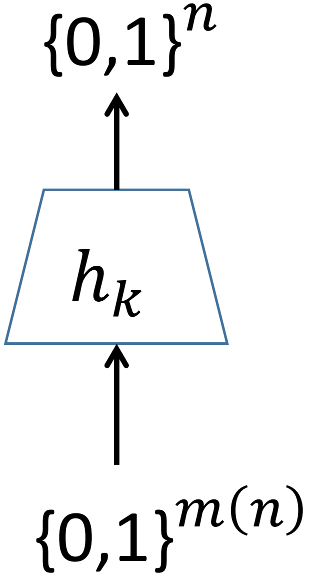
RSA – DSA – EC-DSA…



Hash function families

(Hash) function families

- $H_n := \{h_k: \{0,1\}^{m(n)} \to \{0,1\}^n\}$
- $m(n) \geq n$
- „efficient“



One-wayness

$H_n \coloneqq \{h_k : \{0,1\}^{m(n)} \rightarrow \{0,1\}^n\}$

$h_k \xleftarrow{\$} H_n$
$x \xleftarrow{\$} \{0,1\}^{m(n)}$
$y_c \leftarrow h_k(x)$

Success if $h_k(x^*) = y_c$

y_c, k

$\mathcal{A}$

x^*

Collision resistance

$$H_n \coloneqq \left\{h_k : \{0,1\}^{m(n)} \rightarrow \{0,1\}^n\right\}$$

$$h_k \xleftarrow{\$} H_n$$

Success if

$h_k(x_1^*) = h_k(x_2^*)$ and
$x_1^* \neq x_2^*$

Second-preimage resistance

$H_n \coloneqq \{h_k\colon \{0,1\}^{m(n)} \to \{0,1\}^n\}$

$h_k \xleftarrow{\$} H_n$

$x_c \xleftarrow{\$} \{0,1\}^{m(n)}$

Success if

$h_k(x_c) = h_k(x^*)$ and

$x_c \neq x^*$

Undetectability

$H_n \coloneqq \{h_k : \{0,1\}^{m(n)} \to \{0,1\}^n\}$

$h_k \xleftarrow{\$} H_n$

$b \xleftarrow{\$} \{0,1\}$

If $b = 1$

$\quad x \xleftarrow{\$} \{0,1\}^{m(n)}$

$\quad y_c \leftarrow h_k(x)$

else

$\quad y_c \xleftarrow{\$} \{0,1\}^n$

y_c, k

$\downarrow$

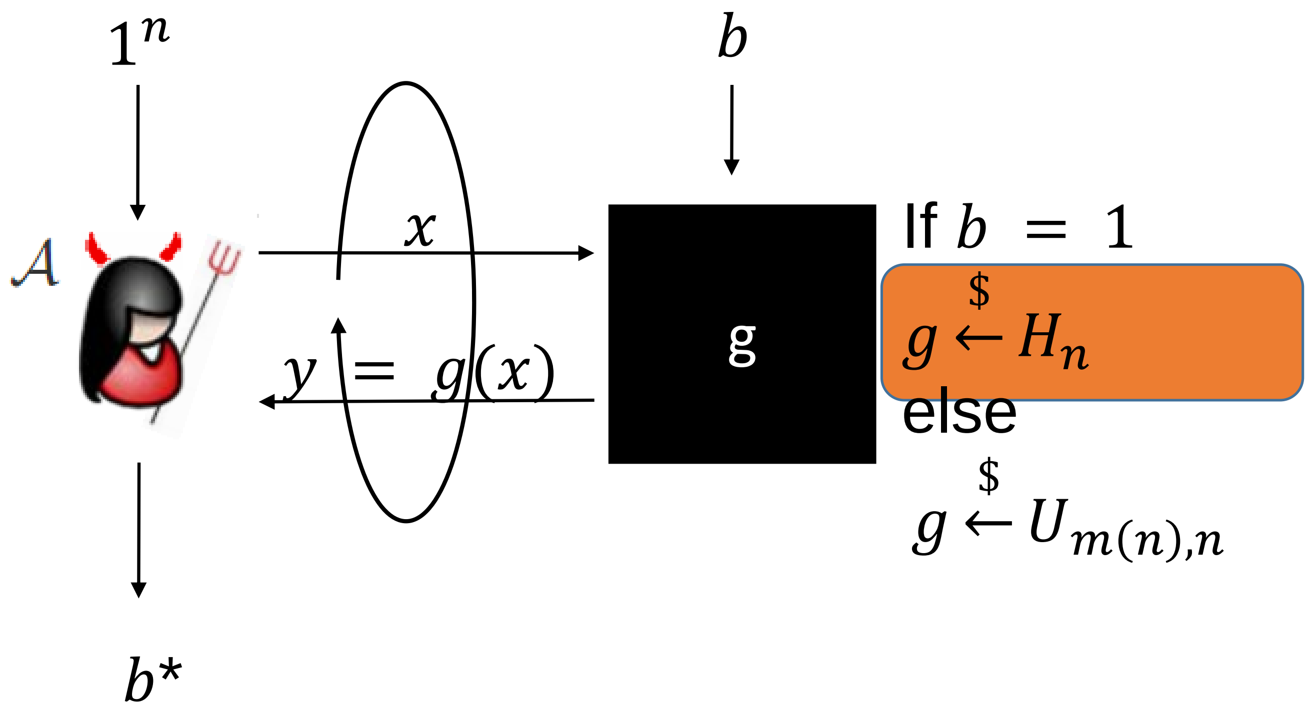
$\mathcal{A}$

$\downarrow$

b^*

Pseudorandomness

$$H_n := \{h_k : \{0,1\}^{m(n)} \rightarrow \{0,1\}^n\}$$



Generic security

- „Black Box“ security (best we can do without looking at internals)
  - For hash functions: Security of random function family
- (Often) expressed in #queries (query complexity)
- Hash functions not meeting generic security considered insecure

Generic Security - OWF

Classically:

- No query: Output random guess

$$Succ_A^{OW} = \frac{1}{2^n}$$

- One query: Guess, check, output new guess

$$Succ_A^{OW} = \frac{2}{2^n}$$

- q-queries: Guess, check, repeat q-times, output new guess

$$Succ_A^{OW} = \frac{q+1}{2^n}$$

- Query bound: $\Theta(2^n)$

Generic Security - OWF

Quantum:

- More complex
- Reduction from quantum search for random H
- Know lower & upper bounds for quantum search!

- Query bound: $\Theta(2^{n/2})$

- Upper bound uses variant of Grover

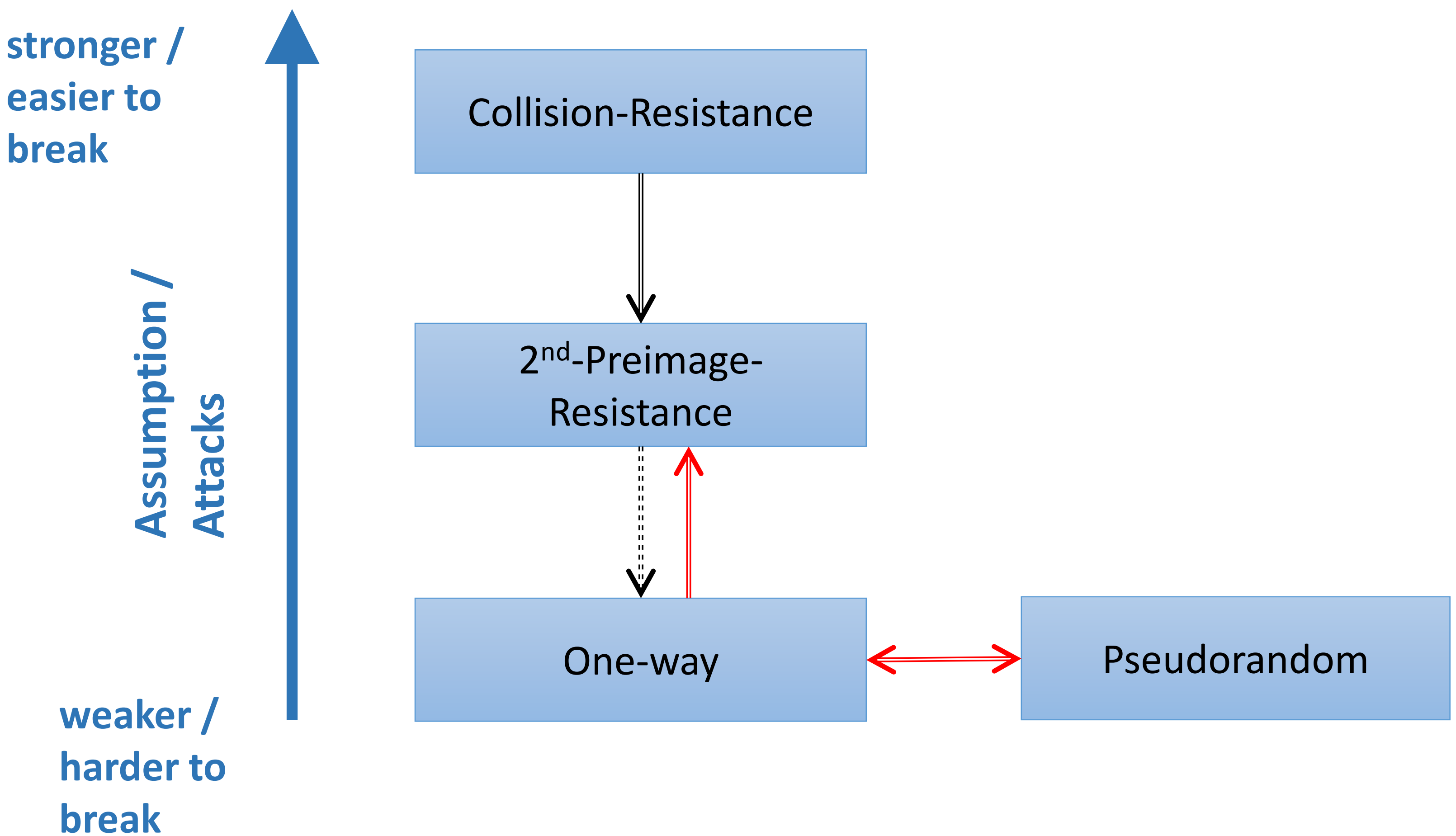
(Disclaimer: Currently only proof for $2^m \gg 2^n$)

Generic Security

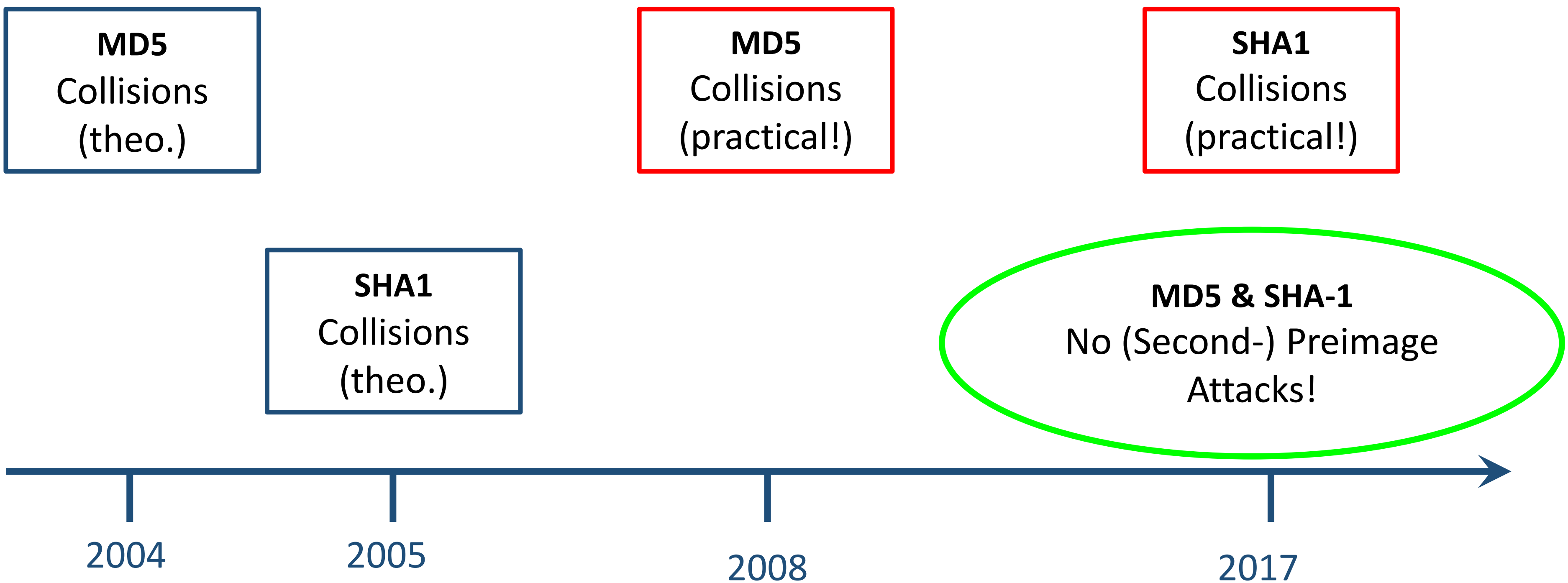
	OW	SPR	CR	UD*	PRF*
Classical	$\Theta(2^n)$	$\Theta(2^n)$	$\Theta(2^{n/2})$	$\Theta(2^n)$	$\Theta(2^n)$
Quantum	$\Theta(2^{n/2})$	$\Theta(2^{n/2})$	$\Theta(2^{n/3})$	$\Theta(2^{n/2})$	$\Theta(2^{n/2})$

* conjectured, no proof

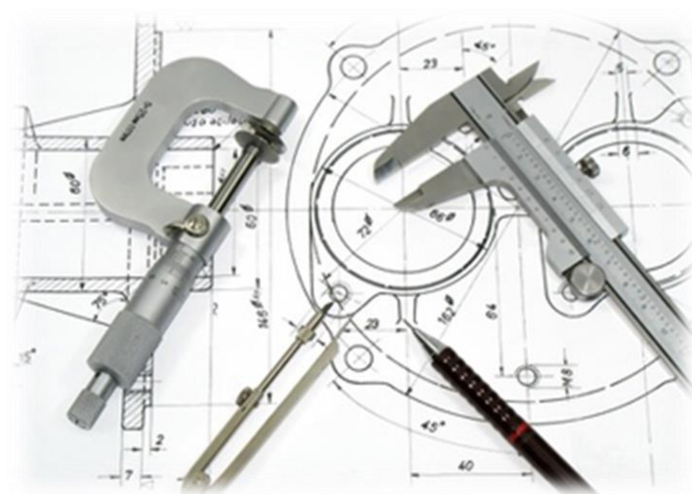
Hash-function properties



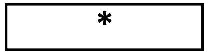
Attacks on Hash Functions

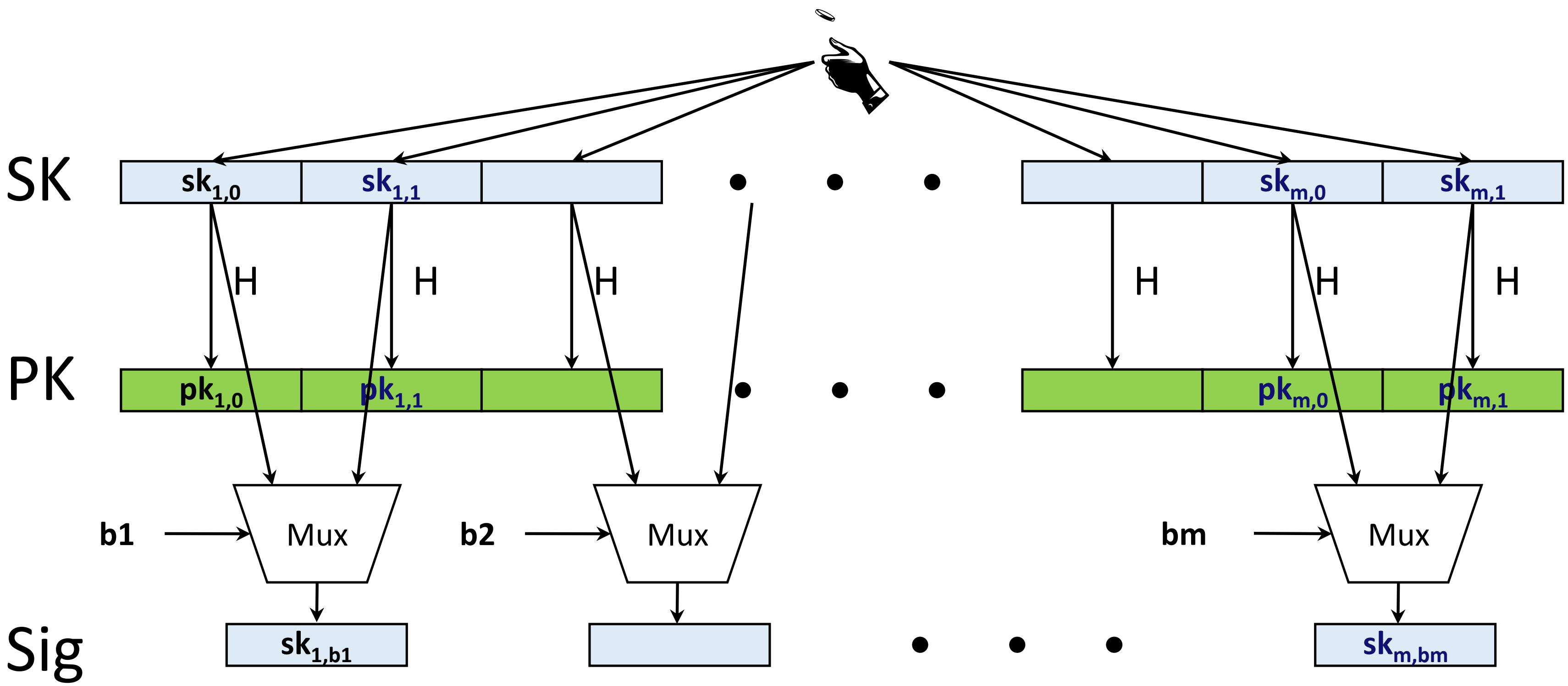


Basic Construction



Lamport-Diffie OTS [Lam79]

Message M = b1,…,bm, OWF H  = n bit

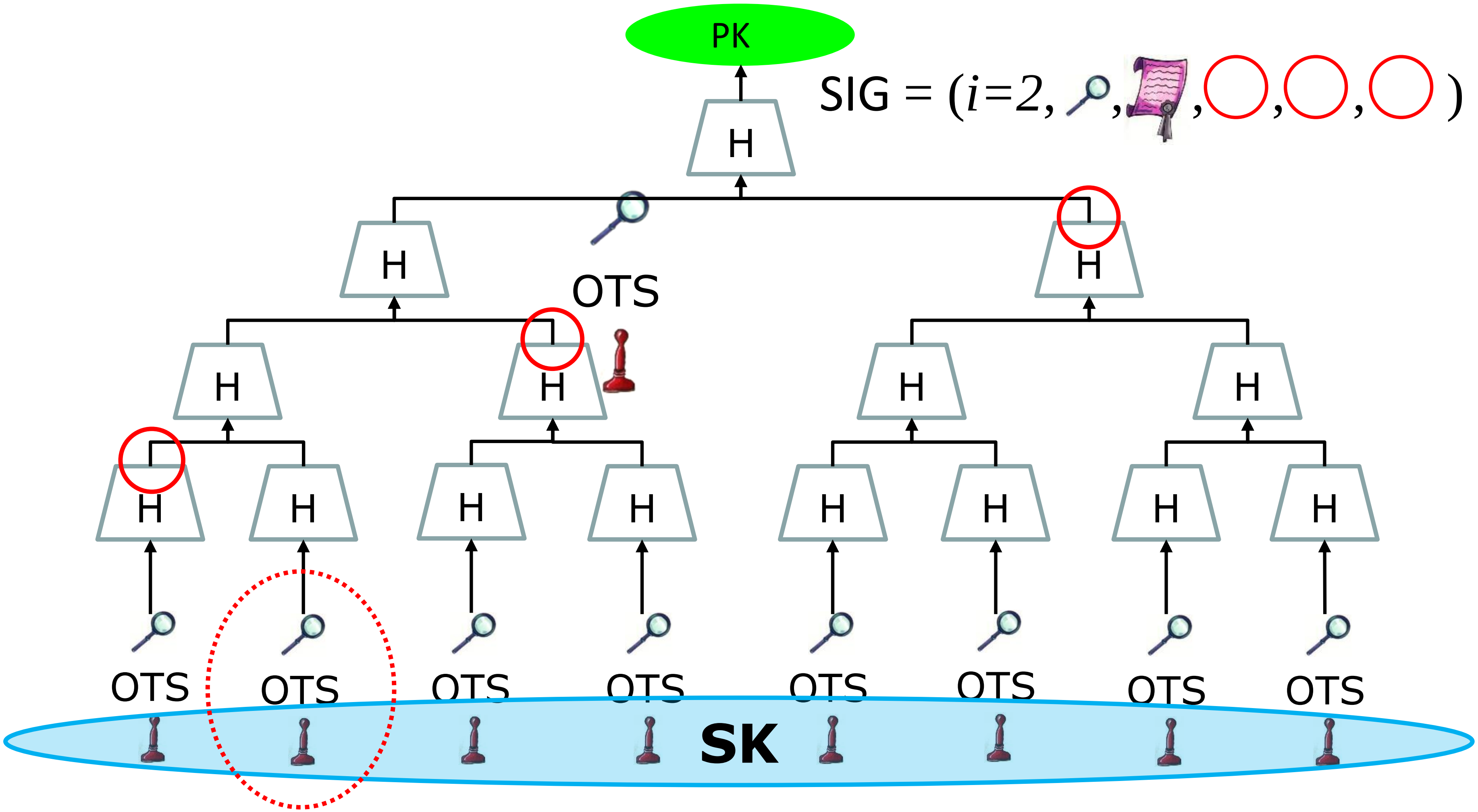


Security

Theorem:

If H is one-way then LD-OTS is one-time eu-cma-secure.

Merkle's Hash-based Signatures



Security

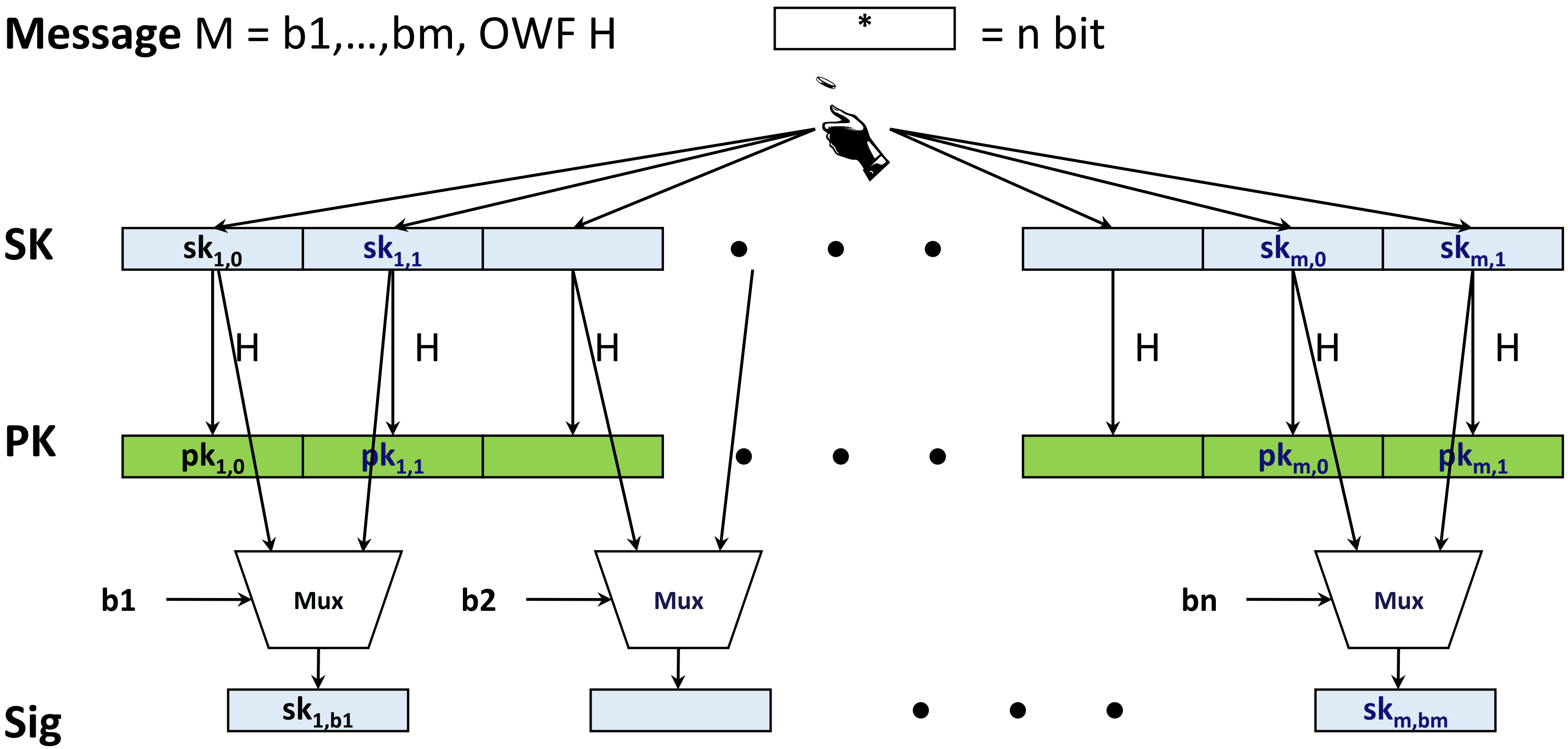
Theorem:

MSS is eu-cma-secure if OTS is a one-time eu-cma secure signature scheme and H is a random element from a family of collision resistant hash functions.

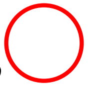
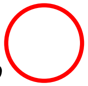
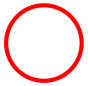
Winternitz-OTS

Recap LD-OTS [Lam79]

Message M = b1,…,bm, OWF H



LD-OTS in MSS

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

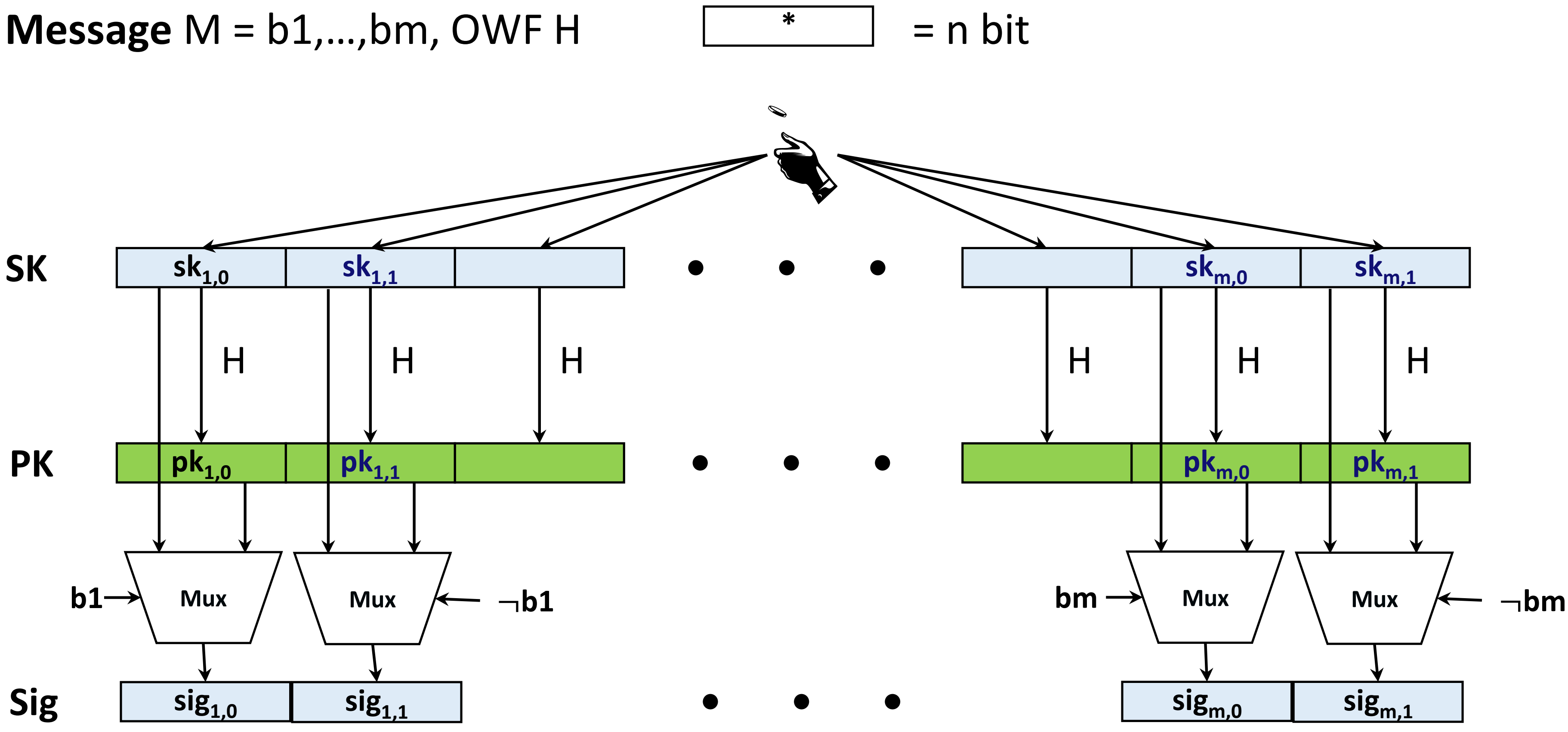
Verification:

1. Verify 
2. Verify authenticity of 

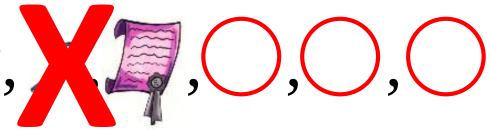
We can do better!

Trivial Optimization

Message M = b1,…,bm, OWF H



Optimized LD-OTS in MSS

SIG = (i=2, )

Verification:

1. Compute  from 
2. Verify authenticity of 

Steps 1 + 2 together verify 

Let's sort this

Message $M = b_1,\ldots,b_m$, OWF H

SK: $sk_1,\ldots,sk_m,sk_{m+1},\ldots,sk_{2m}$

PK: $H(sk_1),\ldots,H(sk_m),H(sk_{m+1}),\ldots,H(sk_{2m})$

Encode M: $M' = M || \neg M$ **=** $b_1,\ldots,b_m, \neg b_1,\ldots,\neg b_m$

(instead of $b_1, \neg b_1,\ldots,b_m,\neg b_m$)

Sig: sig_i **=** $\begin{cases} sk_i & \text{, if } b_i = 1 \\ H(sk_i) & \text{, otherwise} \end{cases}$

Checksum with bad performance!

Optimized LD-OTS

Message M = $b_1,\ldots,b_m$, OWF H

SK: $sk_1,\ldots,sk_m,sk_{m+1},\ldots,sk_{m+1+\log m}$

PK: $H(sk_1),\ldots,H(sk_m),H(sk_{m+1}),\ldots,H(sk_{m+1+\log m})$

Encode M: M' = $b_1,\ldots,b_m,\neg \sum_1^m b_i$

$$\textbf{Sig: } sig_i = \begin{cases} sk_i & \text{, if } b_i = 1 \\ H(sk_i) & \text{, otherwise} \end{cases}$$

IF one b_i is flipped from 1 to 0, another b_j will flip from 0 to 1

Function chains

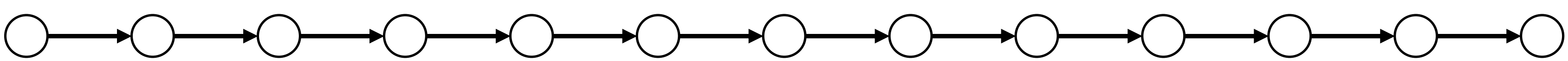
Function family: $H_n := \{h_k : \{0,1\}^n \rightarrow \{0,1\}^n\}$

$h_k \xleftarrow{\$} H_n$

Parameter w

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

$c^0(x) = x$



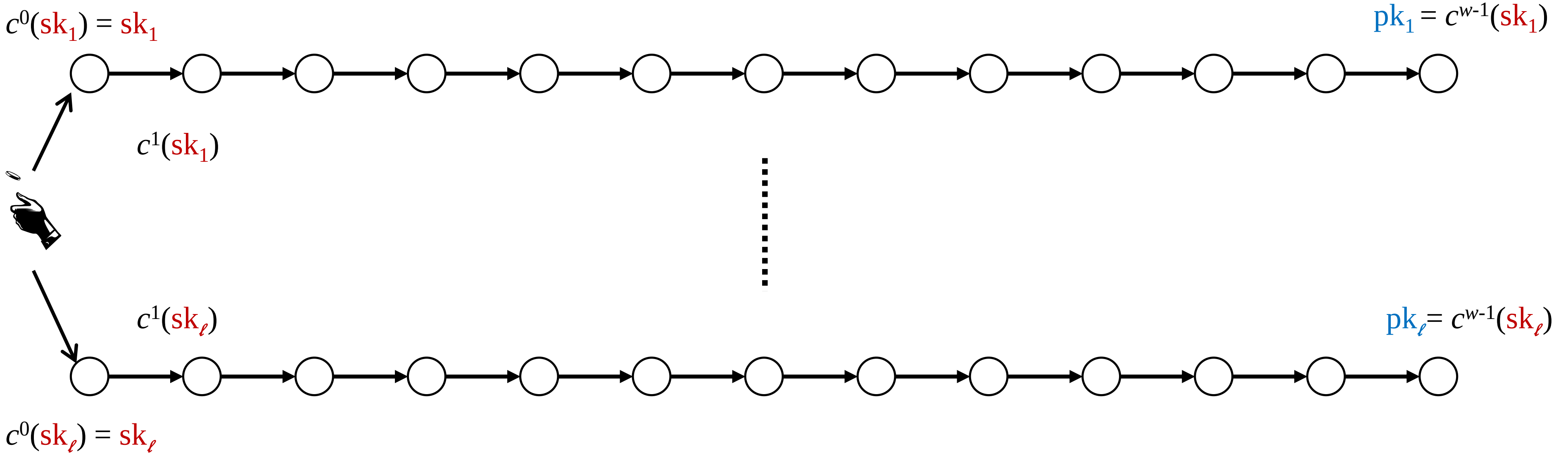
$\boldsymbol{c^{w-1}}(x)$

$c^1(x) = h_k(x)$

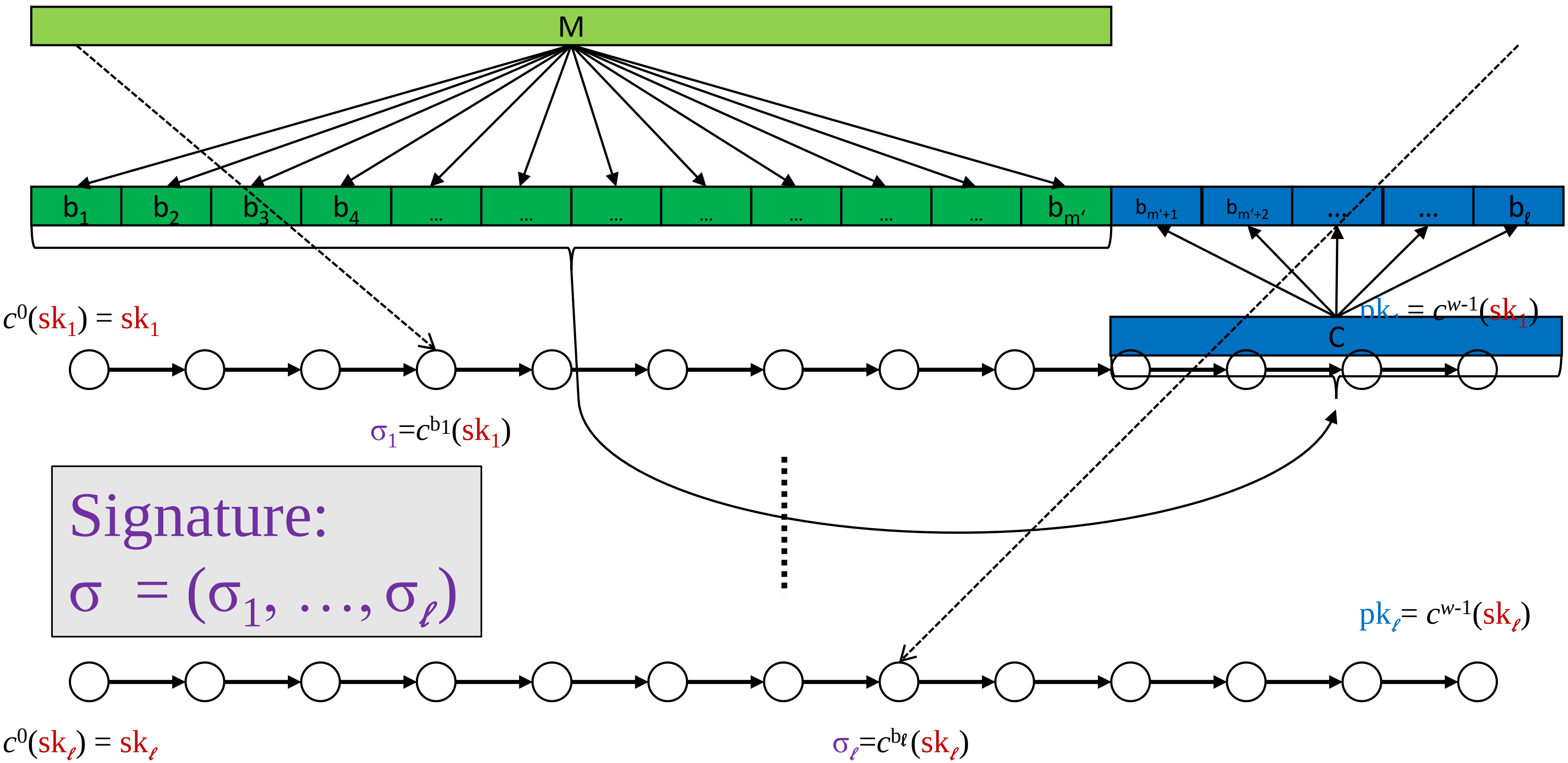
WOTS

Winternitz parameter $\boldsymbol{w}$, security parameter $\boldsymbol{n}$, message length $\boldsymbol{m}$, function family H_n

Key Generation: Compute l, sample h_k

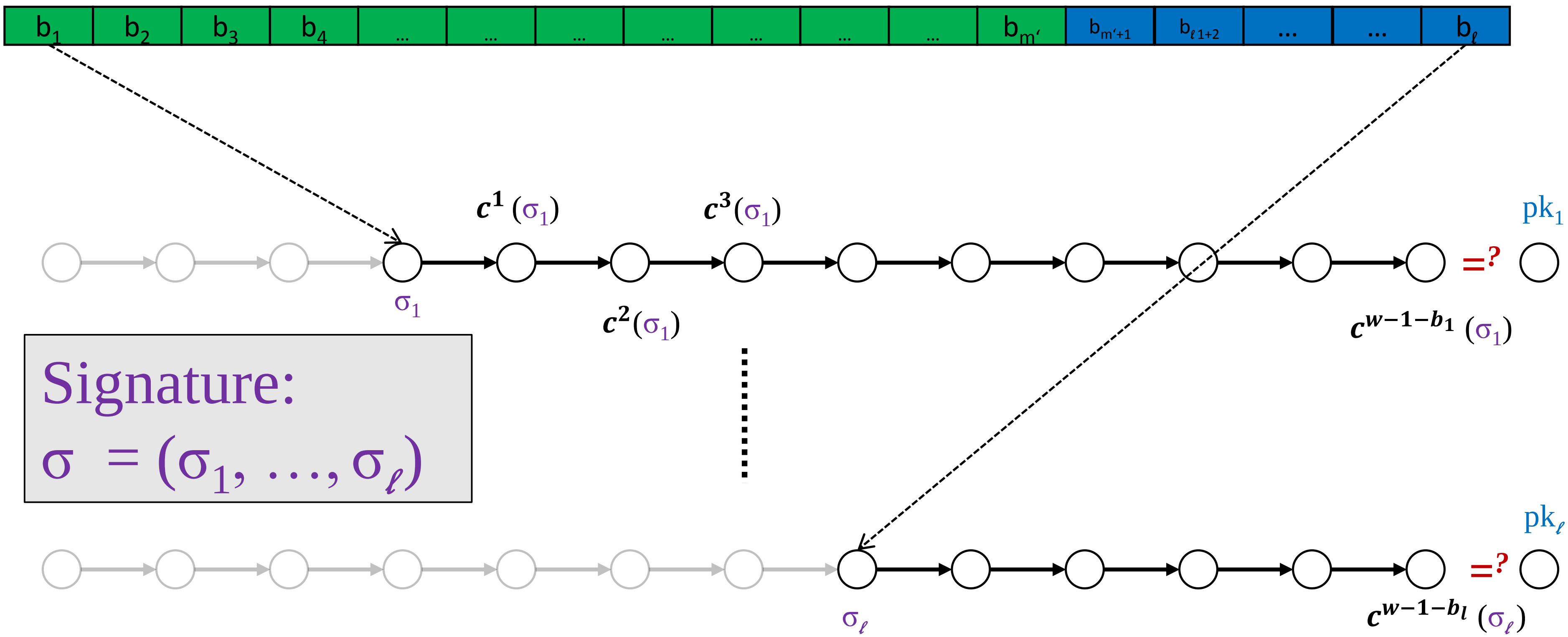


WOTS Signature generation



WOTS Signature Verification

Verifier knows: M, w



WOTS Function Chains

For $x \in \{0,1\}^n$ define $c^0(x) = x$ and

- WOTS: $c^i(x) = h_k(c^{i-1}(x))$
- WOTS$^+$: $c^i(x) = h_k(c^{i-1}(x) \oplus r_i)$

WOTS Security

Theorem (informally):

W-OTS *is strongly unforgeable under chosen message attacks if H_n is a collision resistant family of undetectable one-way functions.*

W-OTS $^{+}$ *is strongly unforgeable under chosen message attacks if H_n is a 2^{nd}-preimage resistant family of undetectable one-way functions.*

XMSS

XMSS

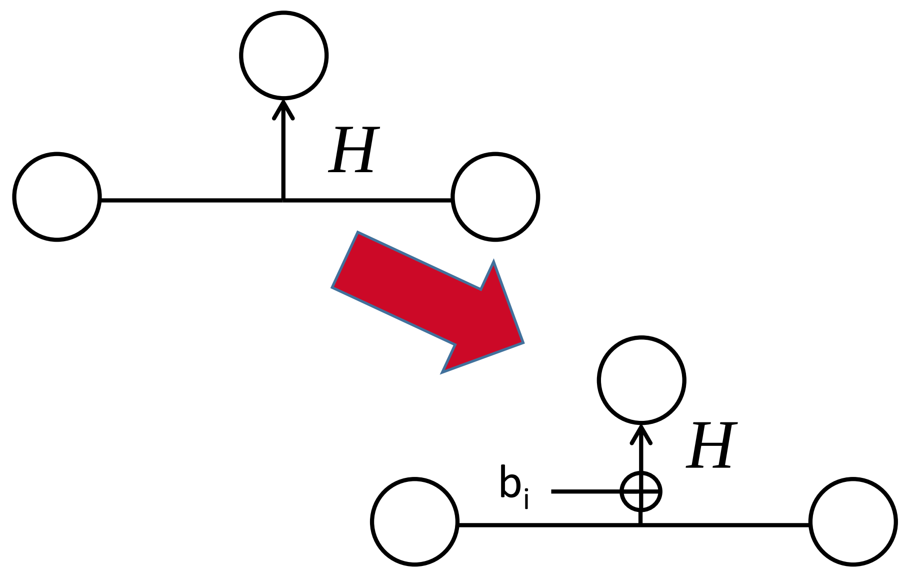
Applies several tricks to achieve **collision-resilience**
-> signature size halved

Tree: Uses bitmasks

Leafs: Use binary tree
with bitmasks

OTS: WOTS$^+$

Mesage digest:
Randomized hashing

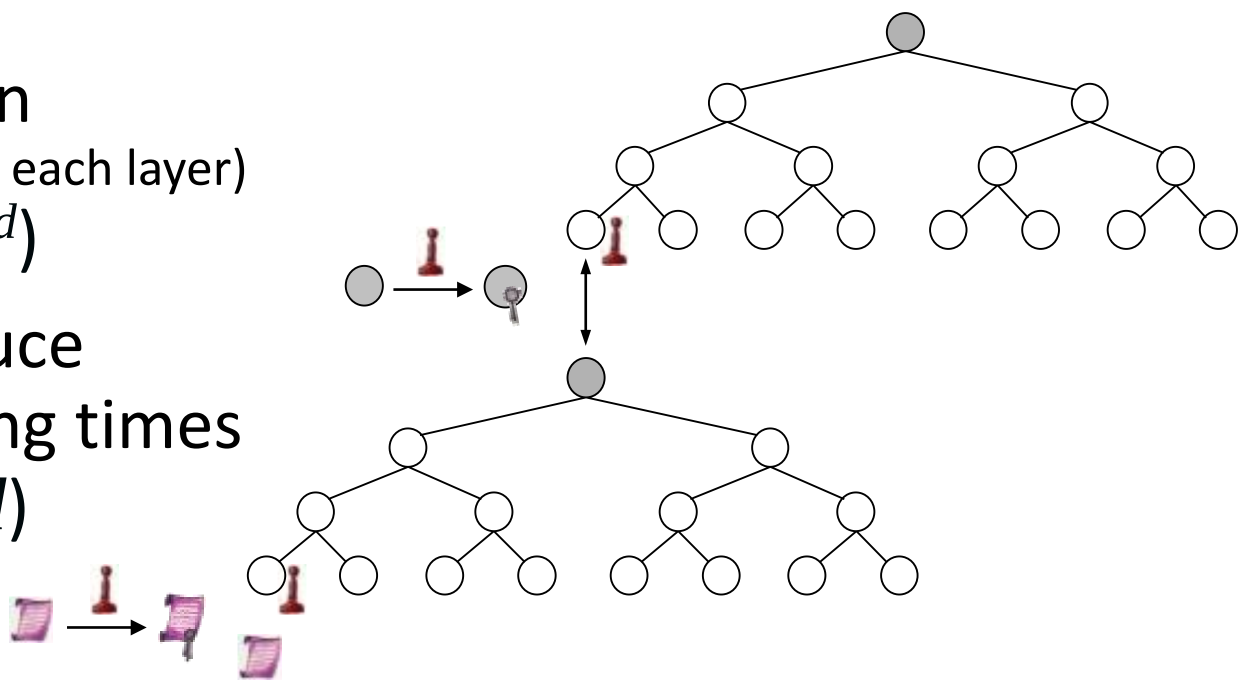


Multi-Tree XMSS

Uses multiple layers of trees to reduce key generation time

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

-> Allows to reduce worst-case signing times
$\Theta(\mathrm{h}/2) \rightarrow \Theta(h/2d)$



XMSS in practice

XMSS Internet-Draft

(draft-irtf-cfrg-xmss-hash-based-signatures)

- Protecting against multi-target attacks / tight security
  - n-bit hash => n bit security
- Small public key (2n bit)
  - At the cost of ROM for proving PK compression secure
- Function families based on SHA2

- Equal to XMSS-T [HRS16] up-to message digest

XMSS / XMSS-T Implementation

C Implementation, using OpenSSL [HRS16]

	Sign (ms)	Signature (kB)	Public Key (kB)	Secret Key (kB)	Bit Security classical/ quantum	Comment
XMSS	3.24	2.8	1.3	2.2	236 / 118	h = 20, d = 1,
XMSS-T	9.48	2.8	**0.064**	2.2	**256 / 128**	h = 20, d = 1
XMSS	3.59	8.3	1.3	14.6	196 / 98	h = 60, d = 3
XMSS-T	10.54	8.3	**0.064**	14.6	**256 / 128**	h = 60, d = 3

Intel(R) Core(TM) i7 CPU @ 3.50GHz
XMSS-T uses message digest from Internet-Draft
All using SHA2-256, w = 16 and k = 2

SPHINCS

About the statefulness

- Works great for some settings

- However....

  ... back-up

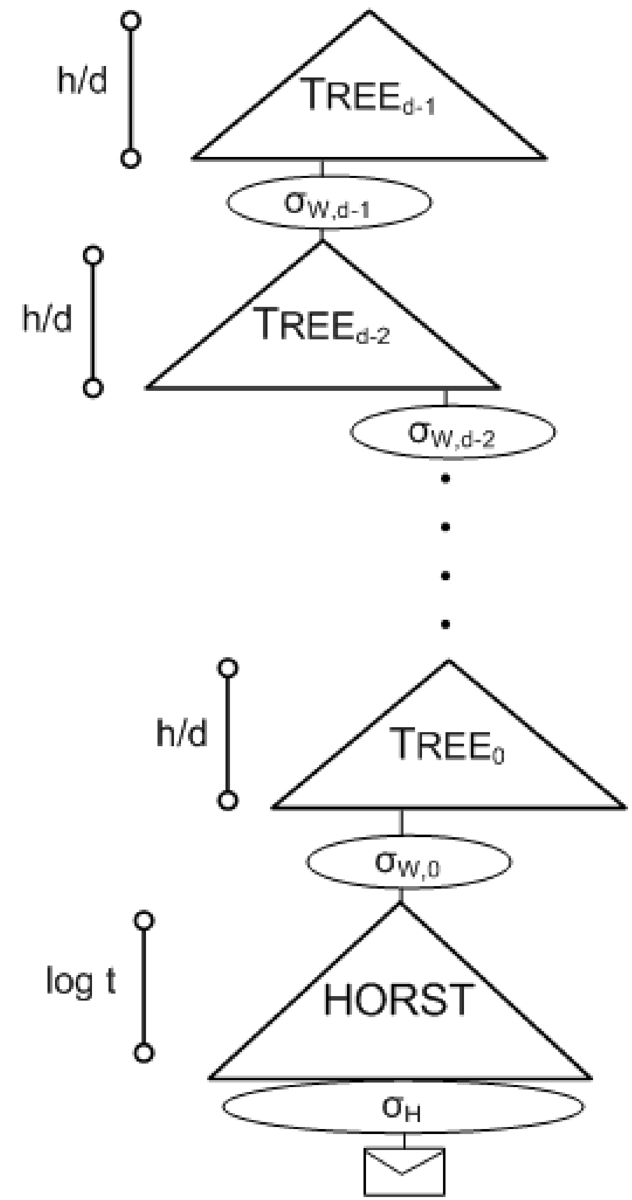
  ... multi-threading

  ... load-balancing

ELIMINATE THE STATE

SPHINCS

- 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



SPHINCS$^+$ (our NIST submission)

- Strengthened security gives smaller signatures
- Collision- and multi-target attack resilient
- Small keys, medium size signatures (lv 3: 17kB)
- THE conservative choice
- No citable speeds yet

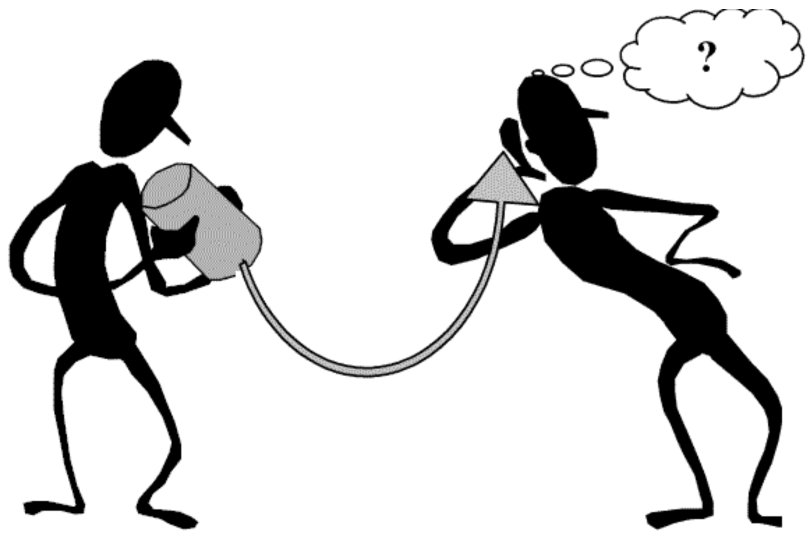
Instantiations

- 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

Thank you!
Questions?



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