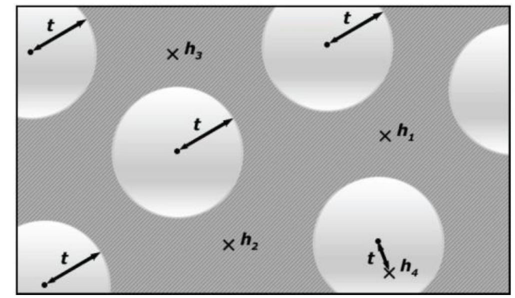
An update on Hash-based Signatures

Andreas Hülsing

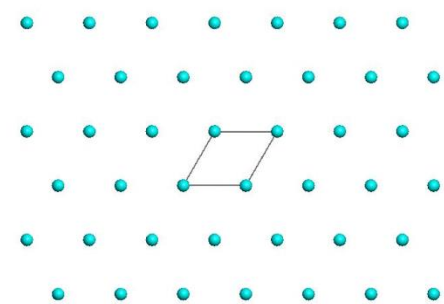
Trapdoor- / Identification Scheme-based (PQ-)Signatures



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

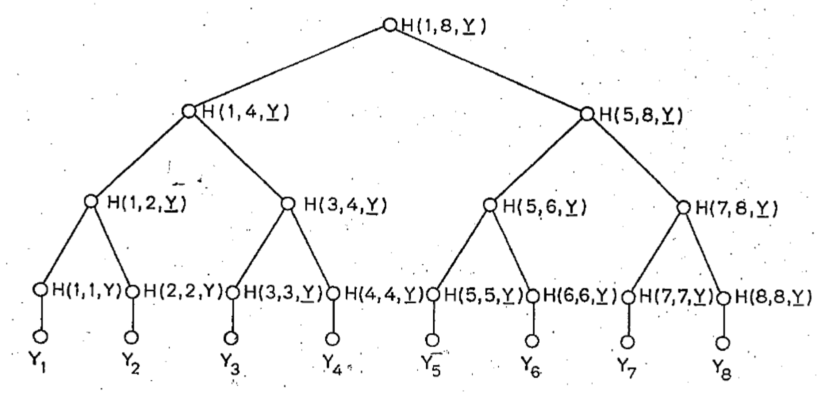
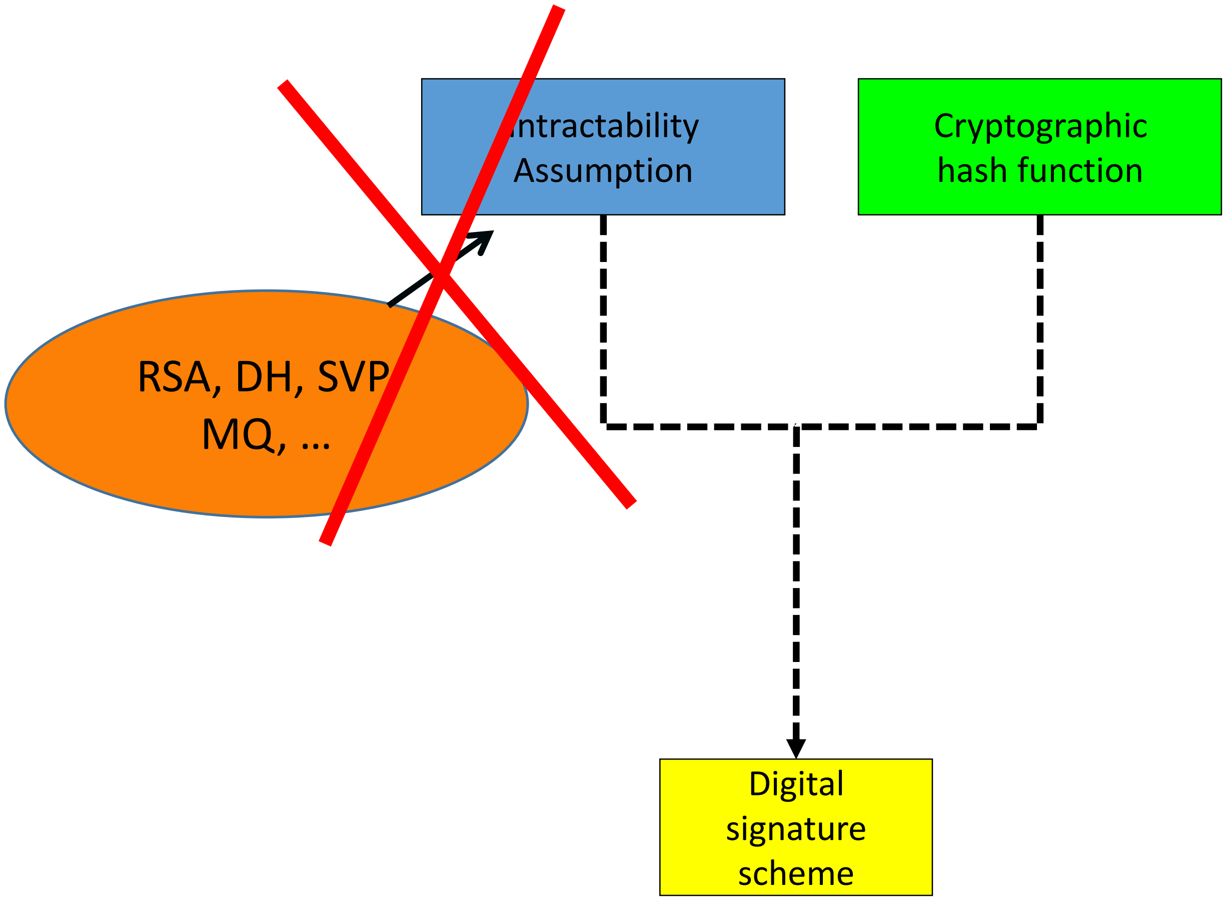
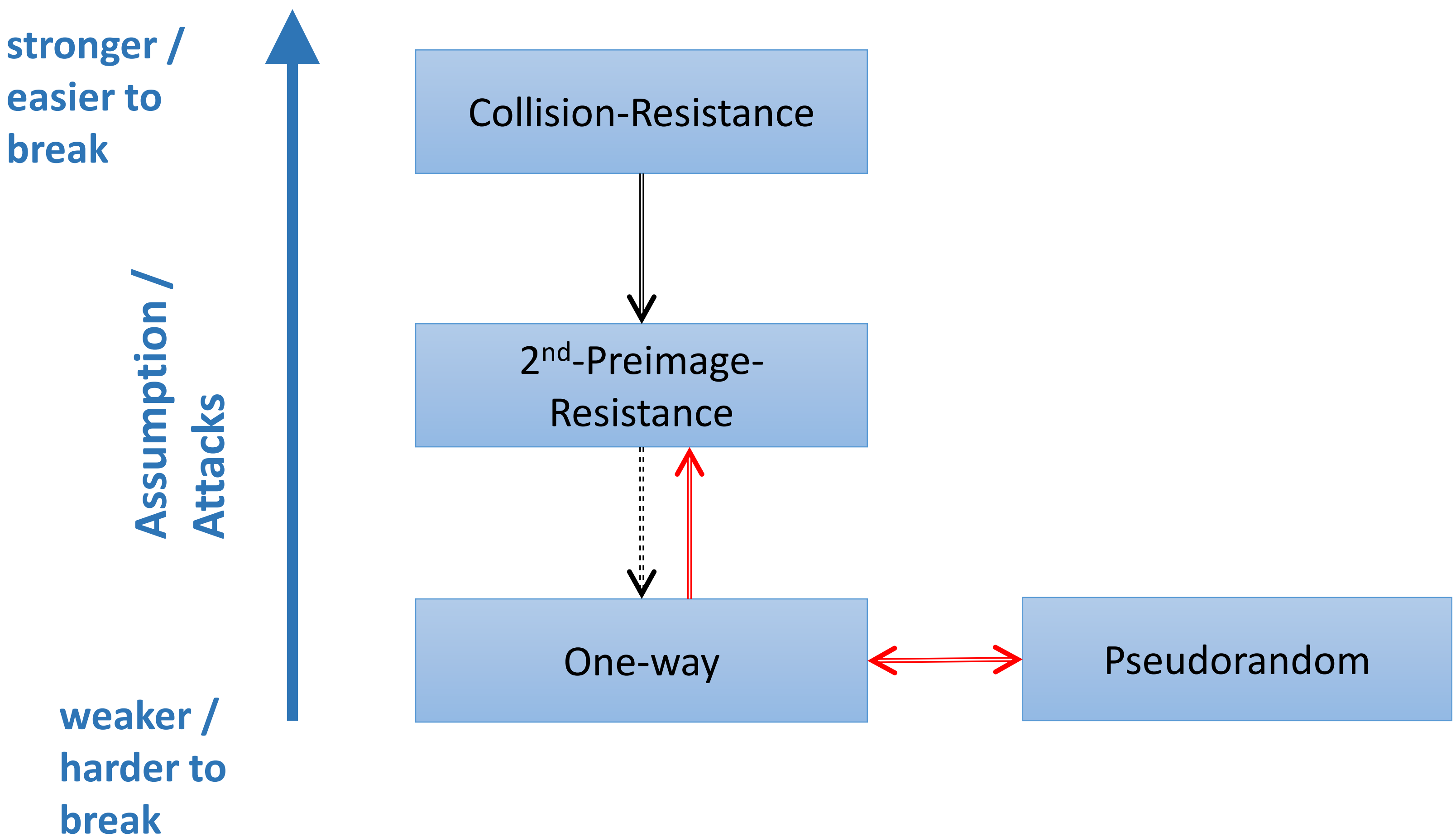


FIG 1
AN AUTHENTICATION TREE WITH N = 8.

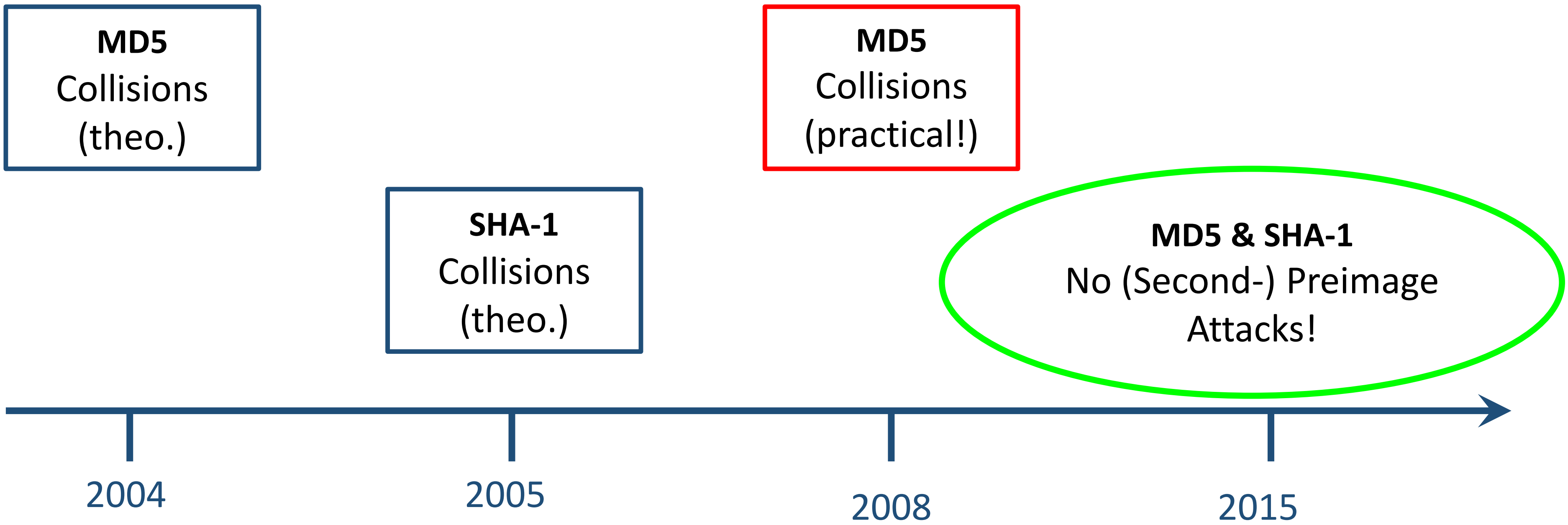
RSA – DSA – EC-DSA…



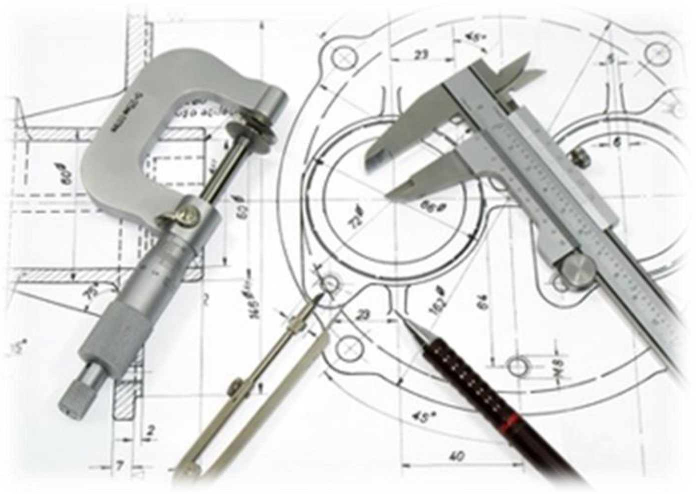
Hash-function properties



Attacks on Hash Functions

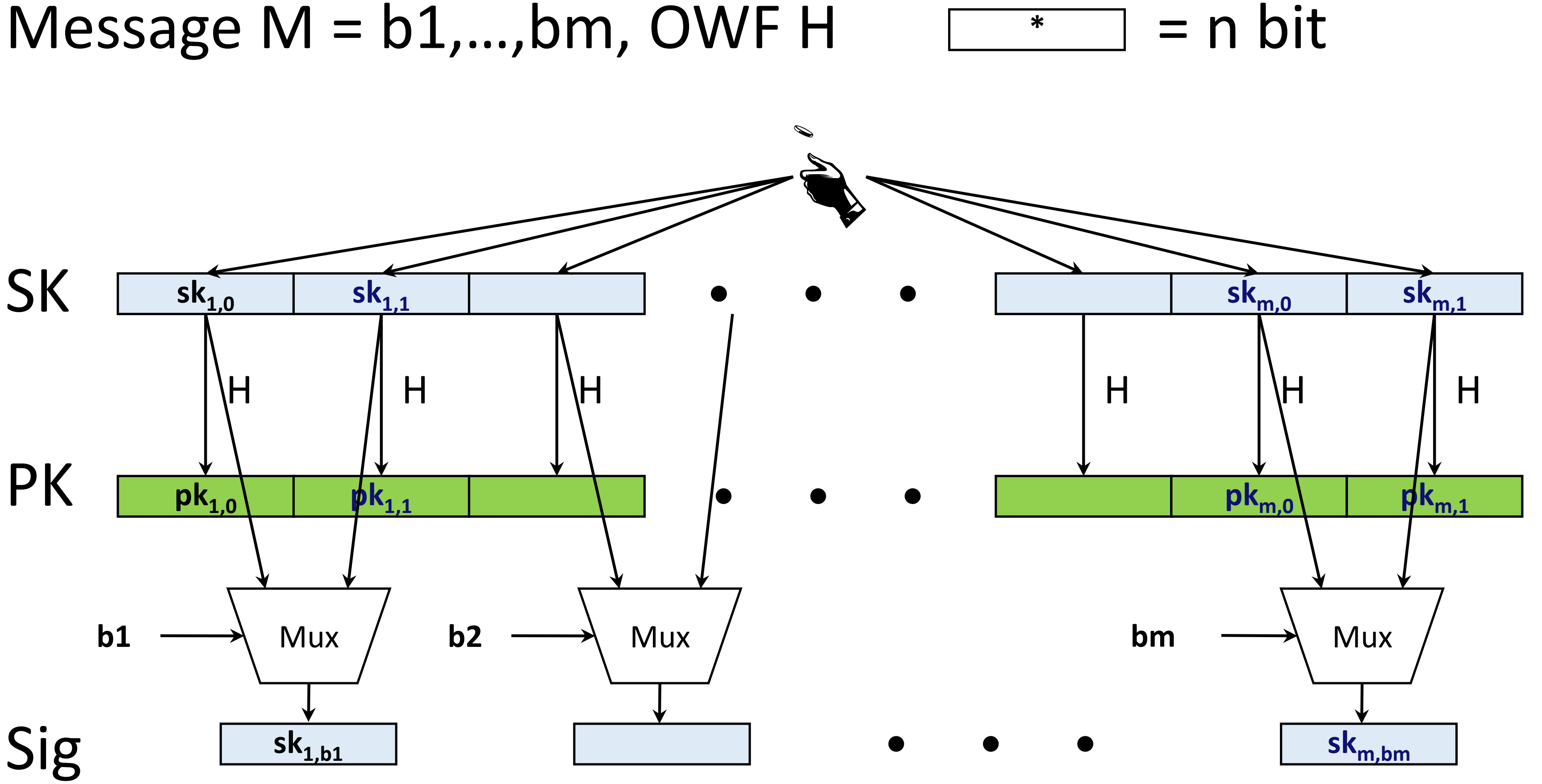


Basic Construction

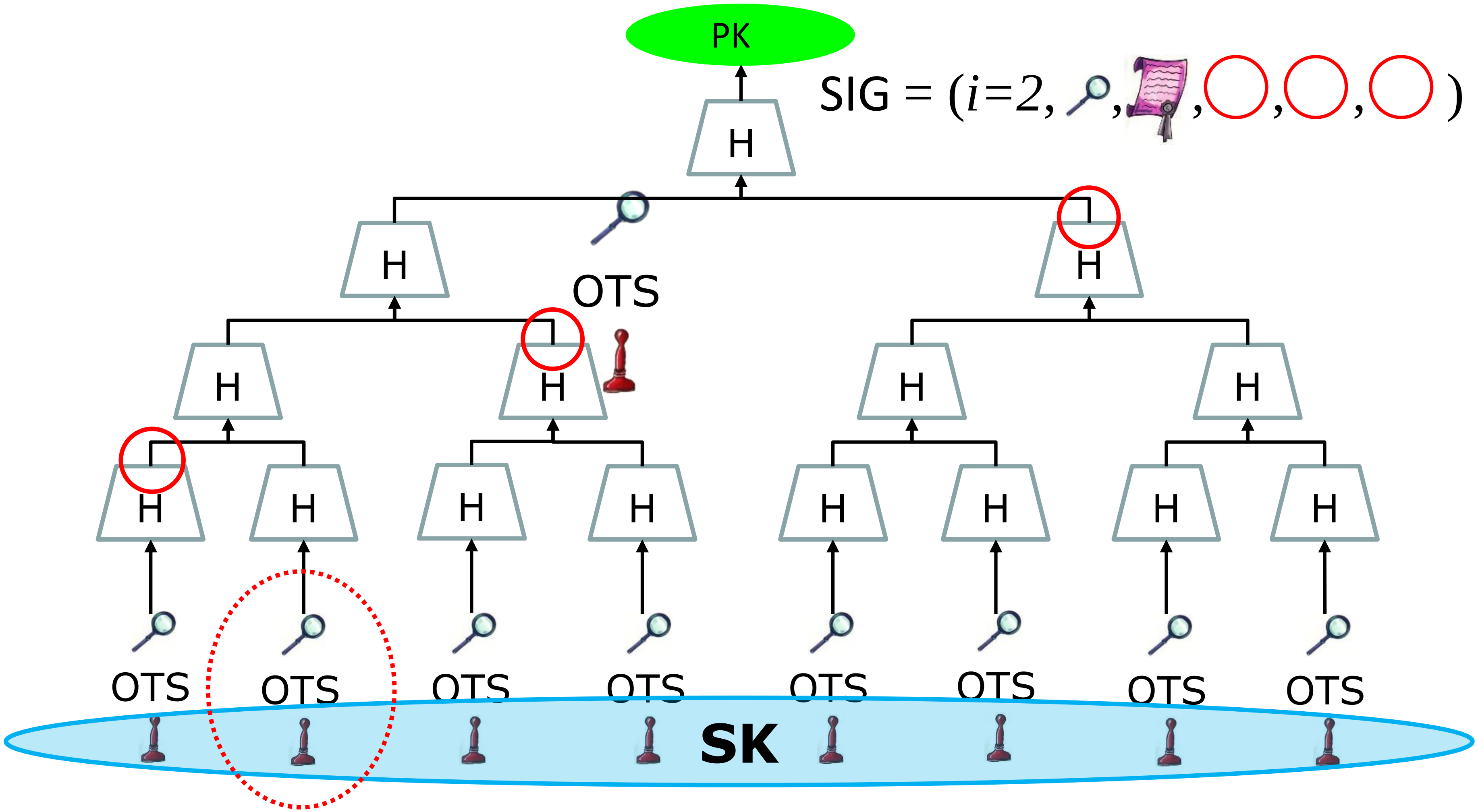


Lamport-Diffie OTS [Lam79]

Message M = b1,…,bm, OWF H



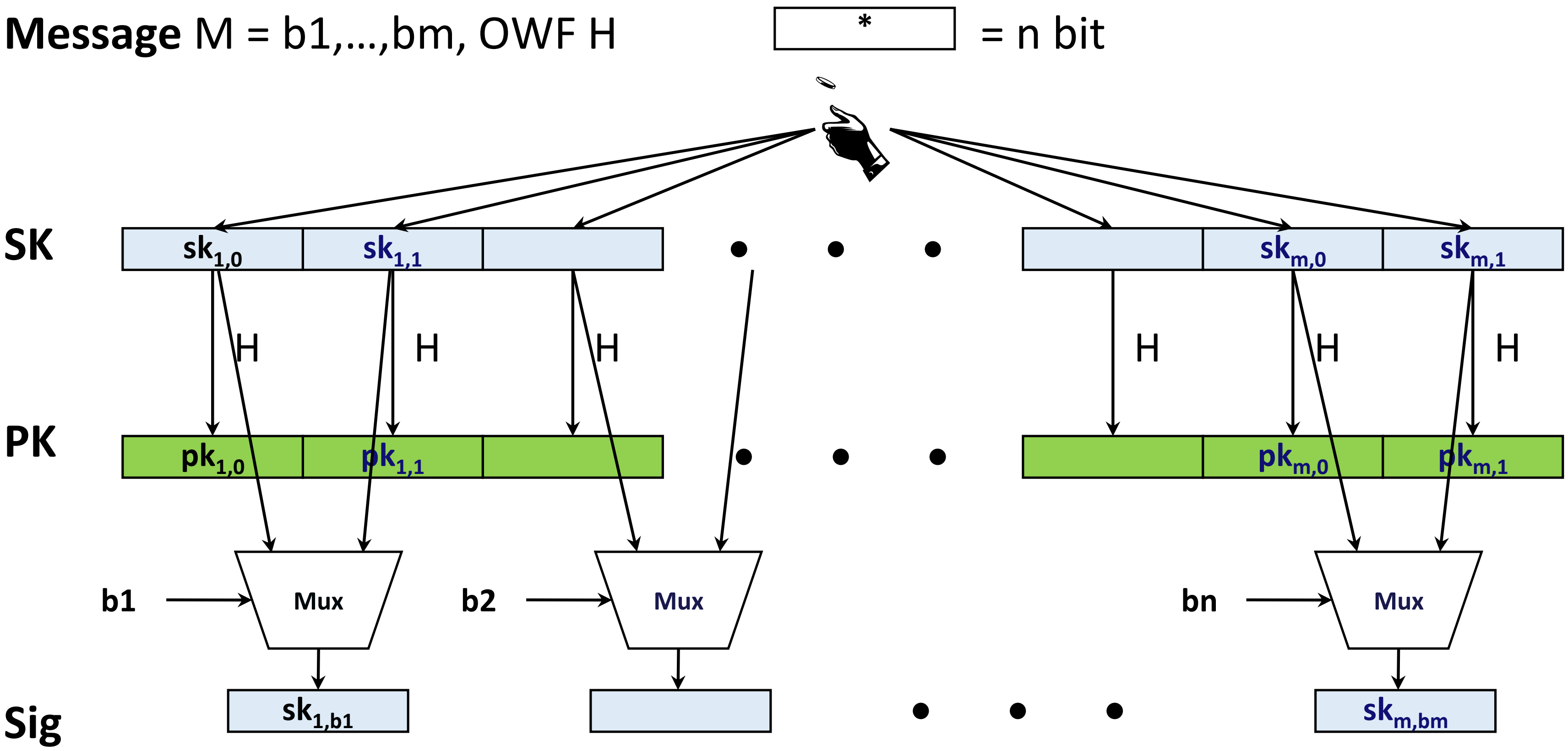
Merkle's Hash-based Signatures



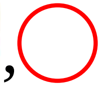
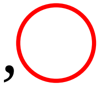
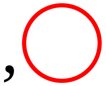
Winternitz-OTS

Recap LD-OTS [Lam79]

Message M = b1,…,bm, OWF H



LD-OTS in MSS

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

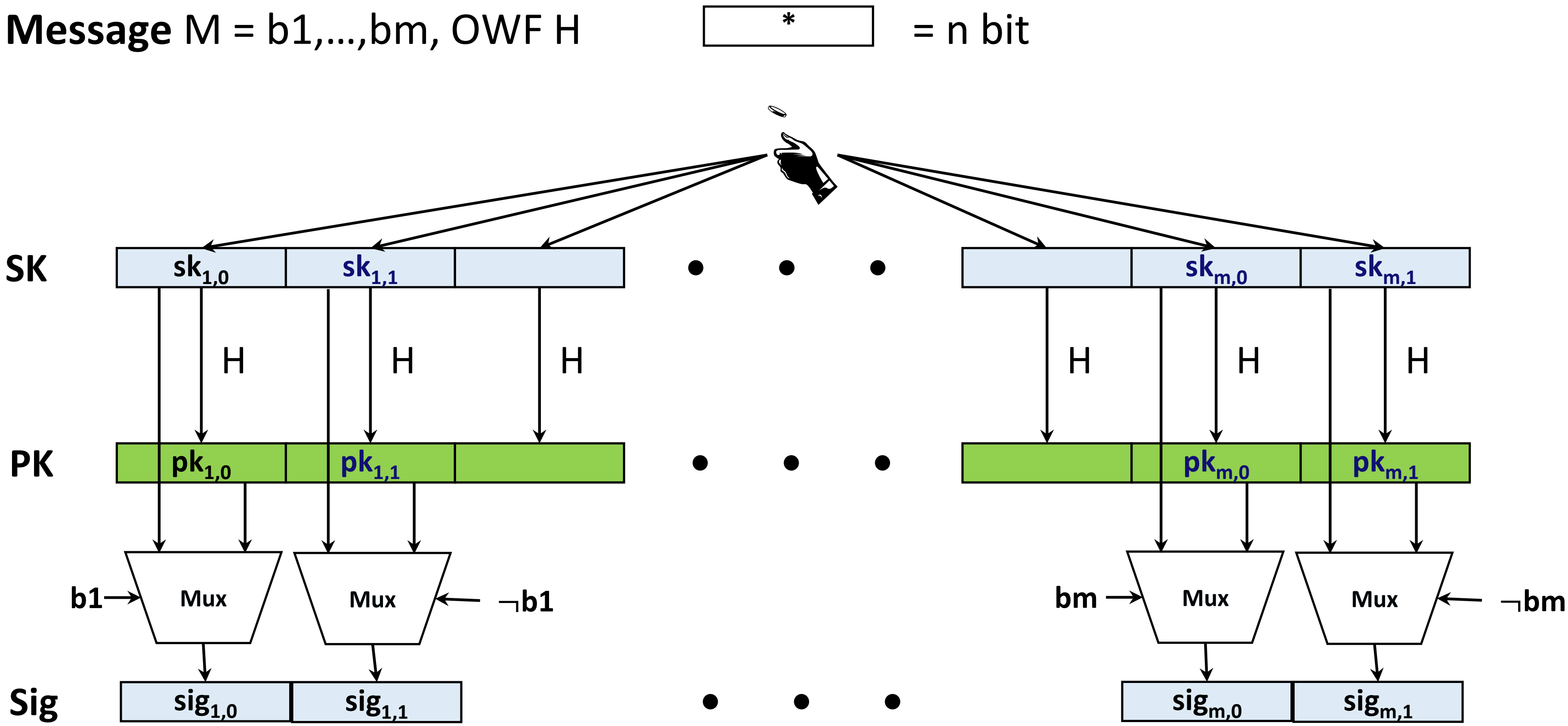
Verification:

1. Verify 
2. Verify authenticity of 

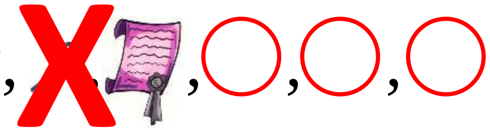
We can do better!

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

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

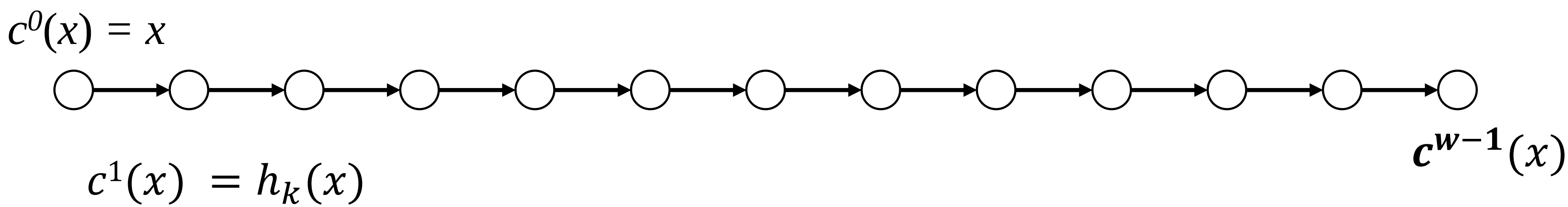
Function chains

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

$h_k \overset{\$}{\leftarrow} H_n$

Parameter w

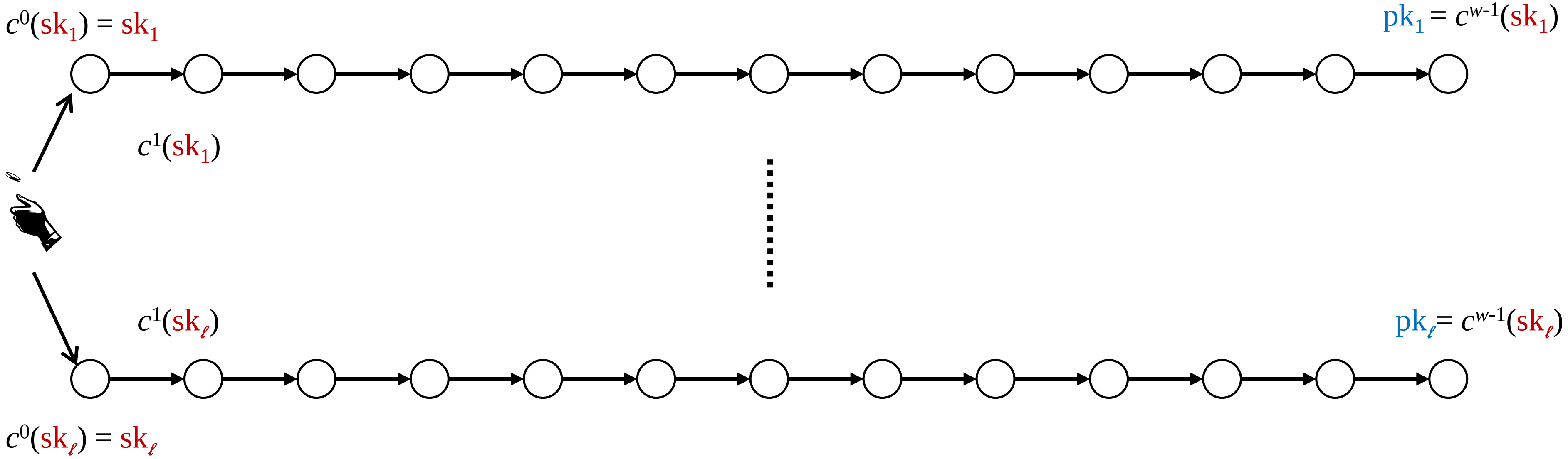
Chain: $c^i(x) = h_k(c^{i-1}(x)) = \underbrace{h_k \circ h_k \circ \ldots \circ h_k(x)}_{i-times}$



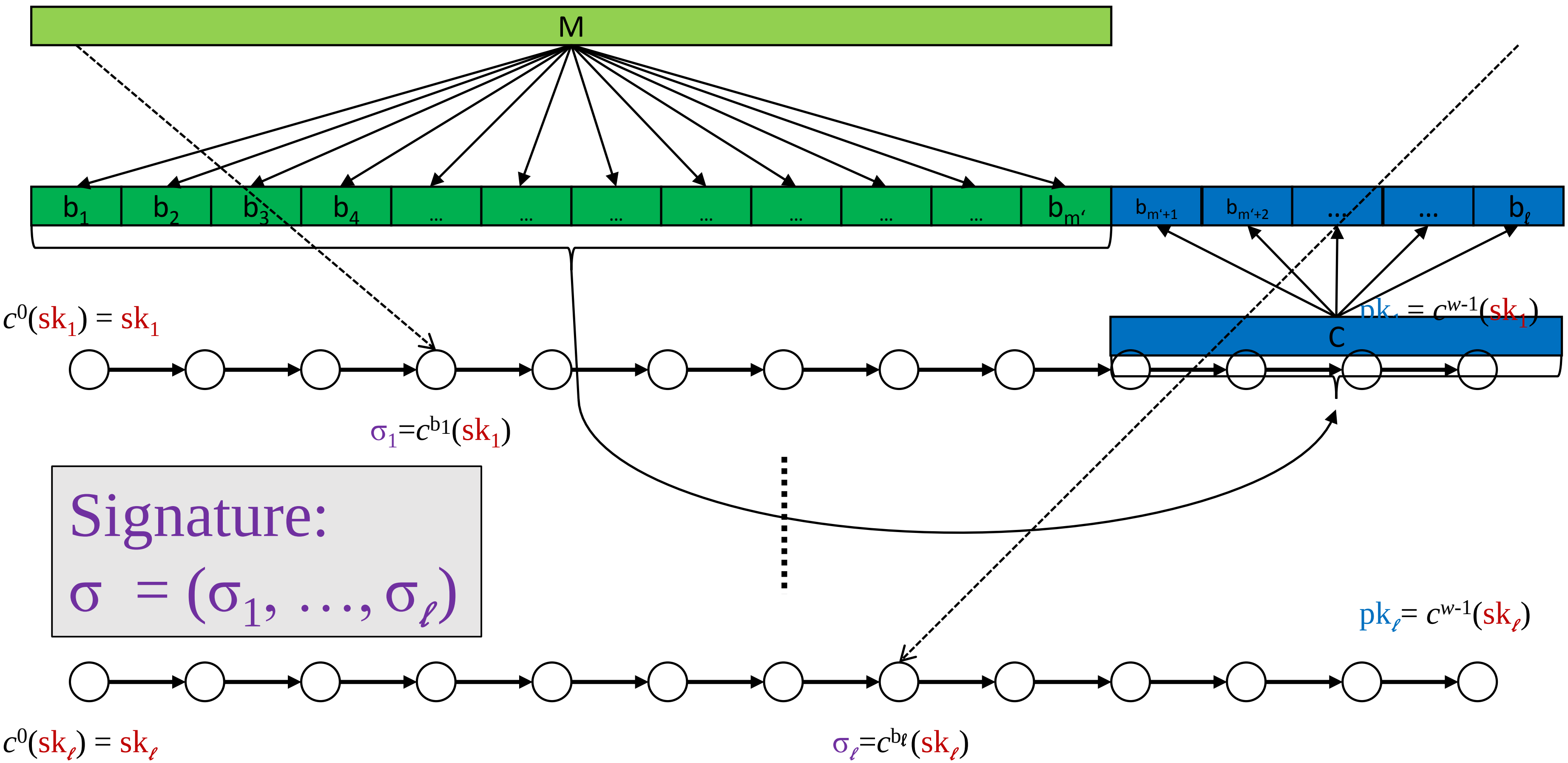
WOTS

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

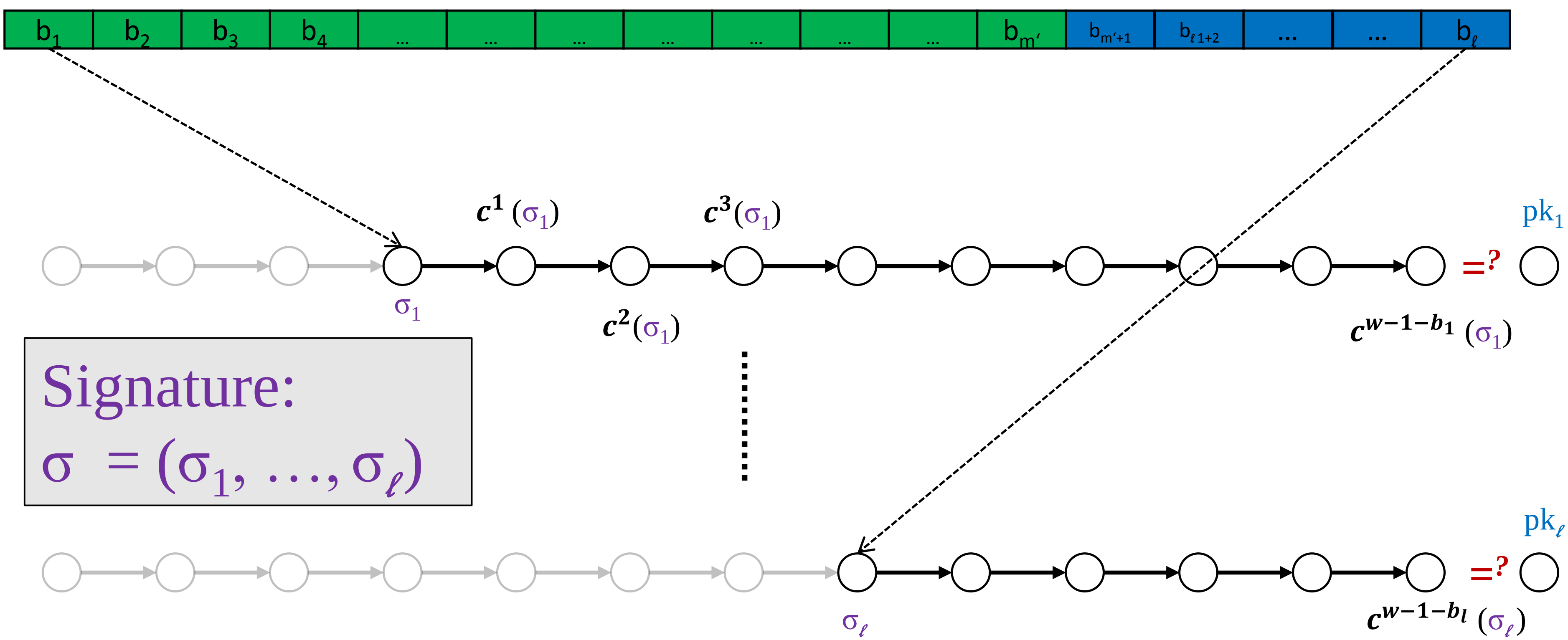


Signature:

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

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$: $c^i(x) = h_{c^{i-1}(x)}(r)$
- 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 existentially unforgeable under chosen message attacks if H_n is a pseudorandom function family.*

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

Standardizing hash-based signatures.
The case of XMSS

XMSS

Tree: Uses bitmasks

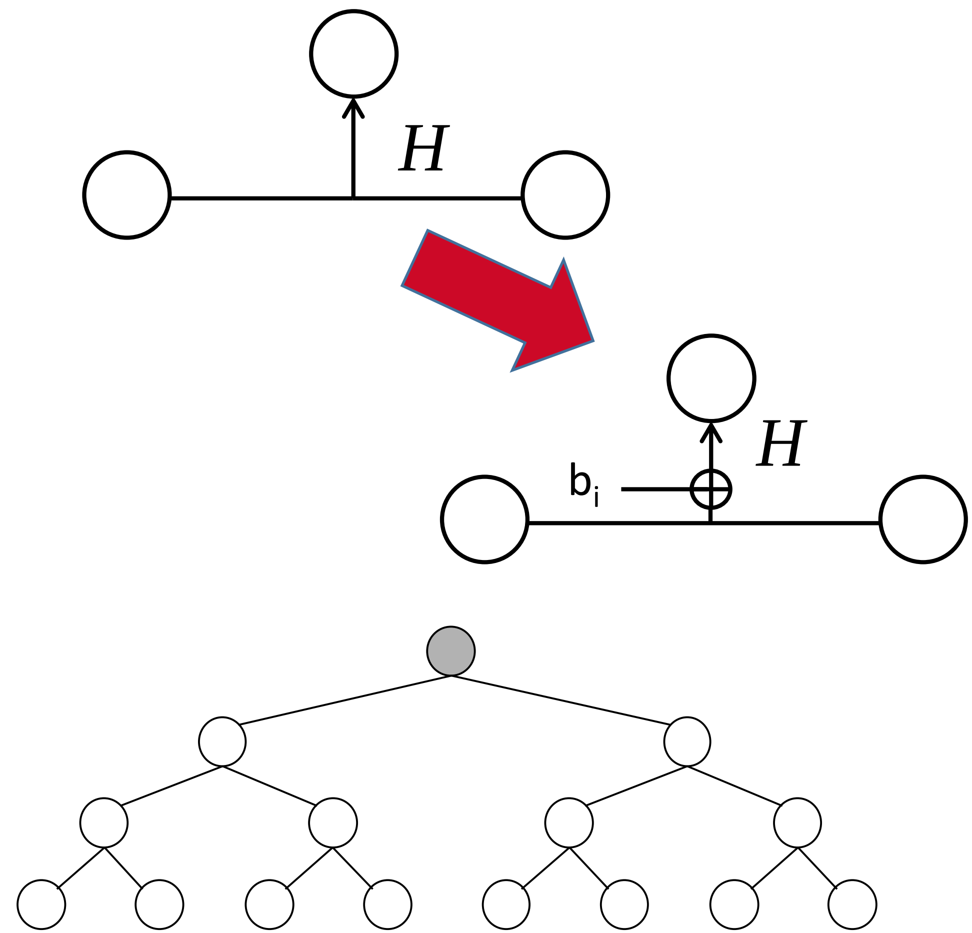
Leafs: Use binary tree with bitmasks

OTS: WOTS$^+$

Mesage digest: Randomized hashing

Collision-resilient

-> signature size halved

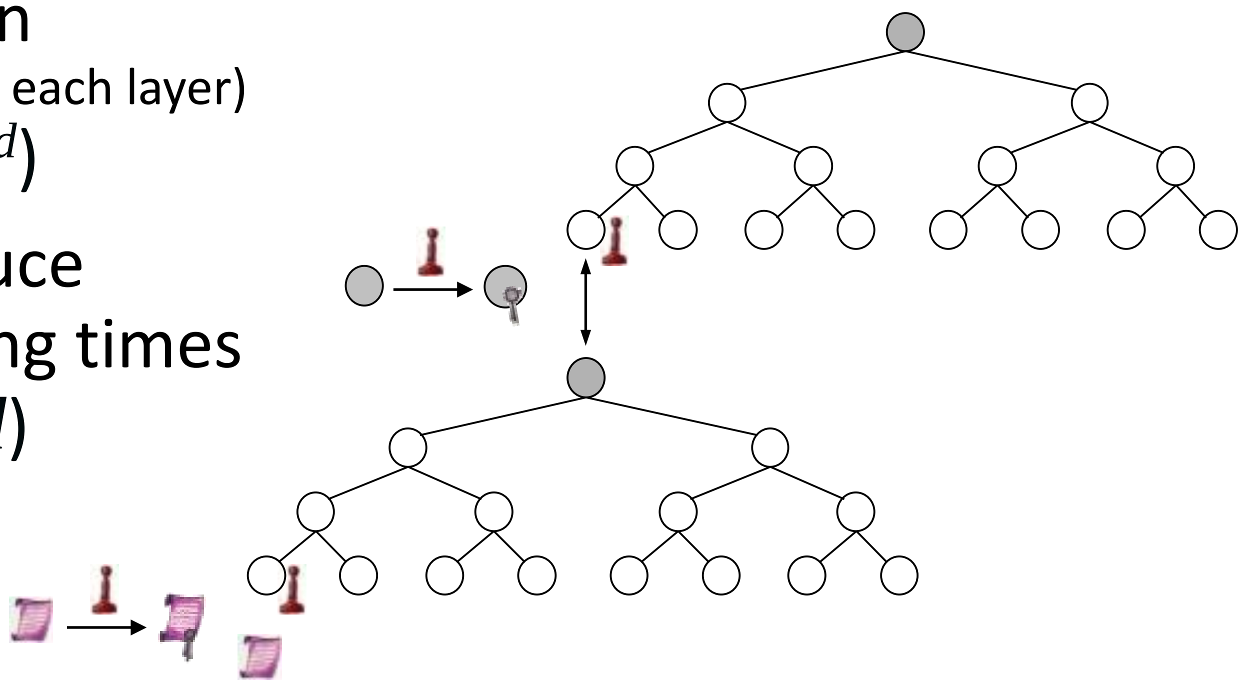


Multi-Tree XMSS

Uses multiple layers of trees

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

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



Multi-target attacks

What is the bit security of XMSS using a n = 256 bit hash function?

256 bit?

No!

Multi-target attacks

It suffices to invert h_k on one out of

$$\sim N \cdot w \cdot l$$

different values. (For N= #WOTS key pairs, m = message length, w = Winternitz parameter, l = |WOTS message encoding|)

Attack complexity: $2^{n - \log(Nwl)}$

For $\mathrm{n} = \mathrm{m} = 256, N = 2^{20}, w = 16, l \sim 64$

approx. 226 bit security

Similar problem applies for second-preimage resistance.

Multi-target attacks

Attack complexity: $2^{n - \log(Nwl)}$

Reason:

- Many targets for same function
- Each hash query can be used for all targets
- Dependent problems

Solution?

Use different elements from function family for each hash (and different bitmasks).

- Makes problems independent
- Each hash query can only be used for one target!

XMSS-Draft since -01

Each hash function call (excl. message hash) takes now a key and a bitmask.

Issue: Order of $N \cdot w \cdot l$ keys and bitmasks that have to be published.

Put them into PK? Impractical

Solution: PRG + Seed in PK

XMSS-Draft since -01

Solution: PRG + Seed in PK

Security:

- Not really standard model.
- Natural but new assumption („Generating the public values using a PRG, the scheme does not get less secure if seed is published.“),
- Or ROM

SPHINCS: practical stateless hash-based signatures

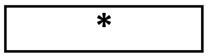
joint work with Daniel J. Bernstein, Daira Hopwood, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, Zooko Wilcox O’Hearn

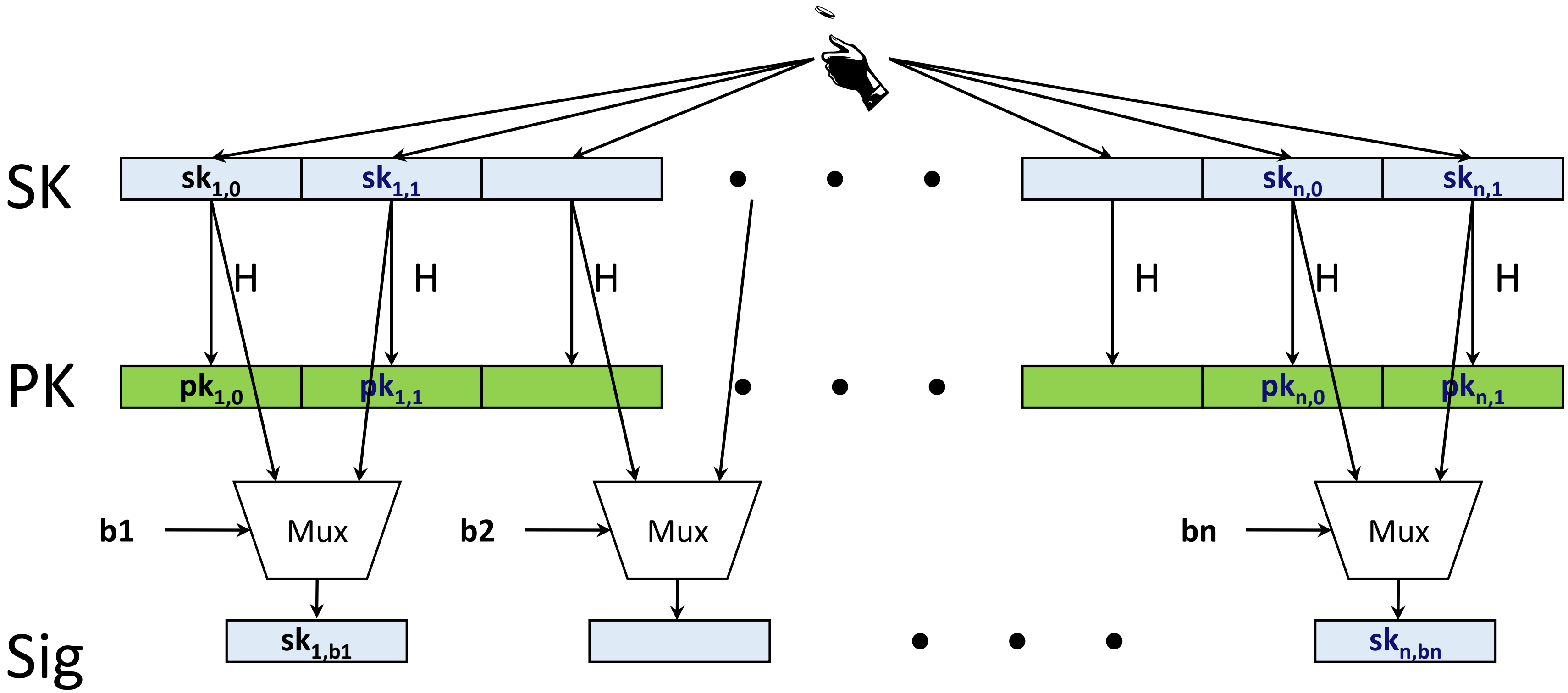
ELIMINATE THE STATE

Protest?

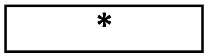
Few-Time Signature Schemes

Recap LD-OTS

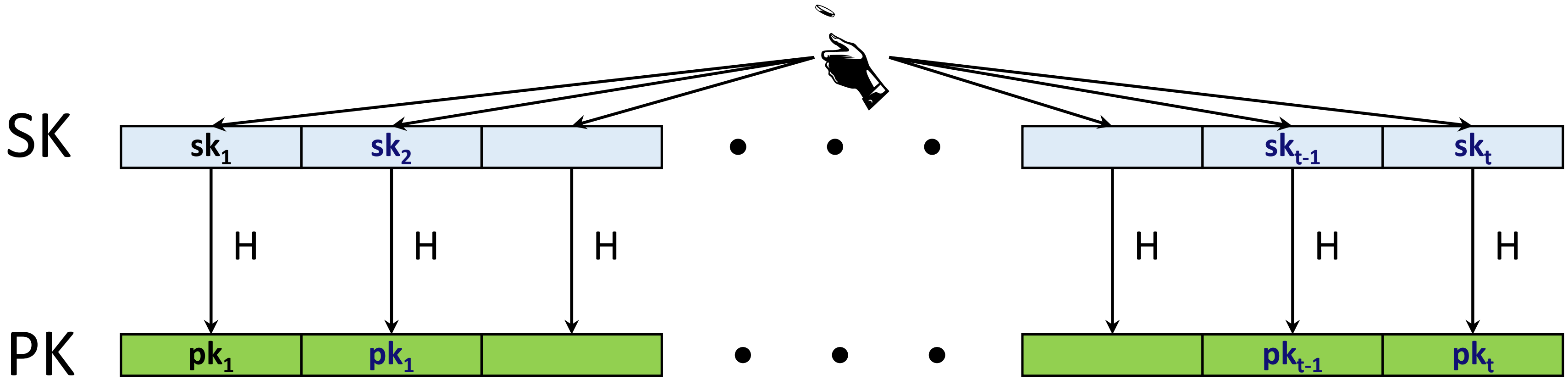
Message M = b1,…,bn, OWF H  = n bit



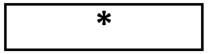
HORS [RR02]

Message M, OWF H, CRHF H’  = n bit

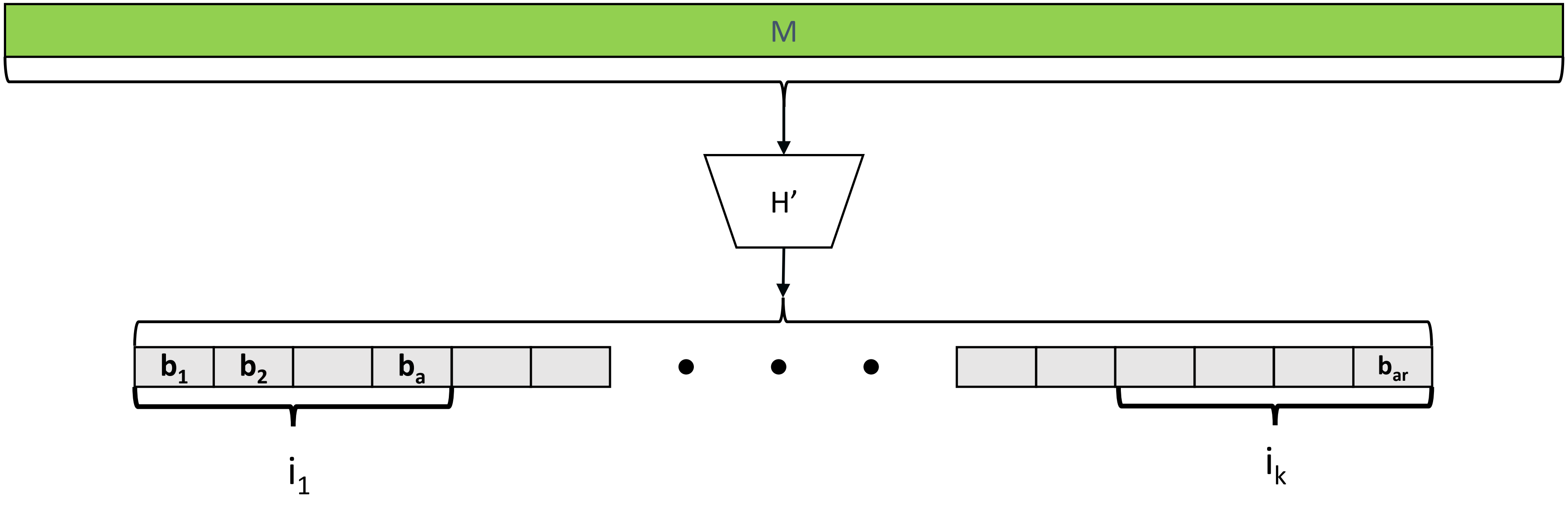
Parameters $t=2^a$,k, with m = ka (typical a=16, k=32)



HORS mapping function

Message M, OWF H, CRHF H’  = n bit

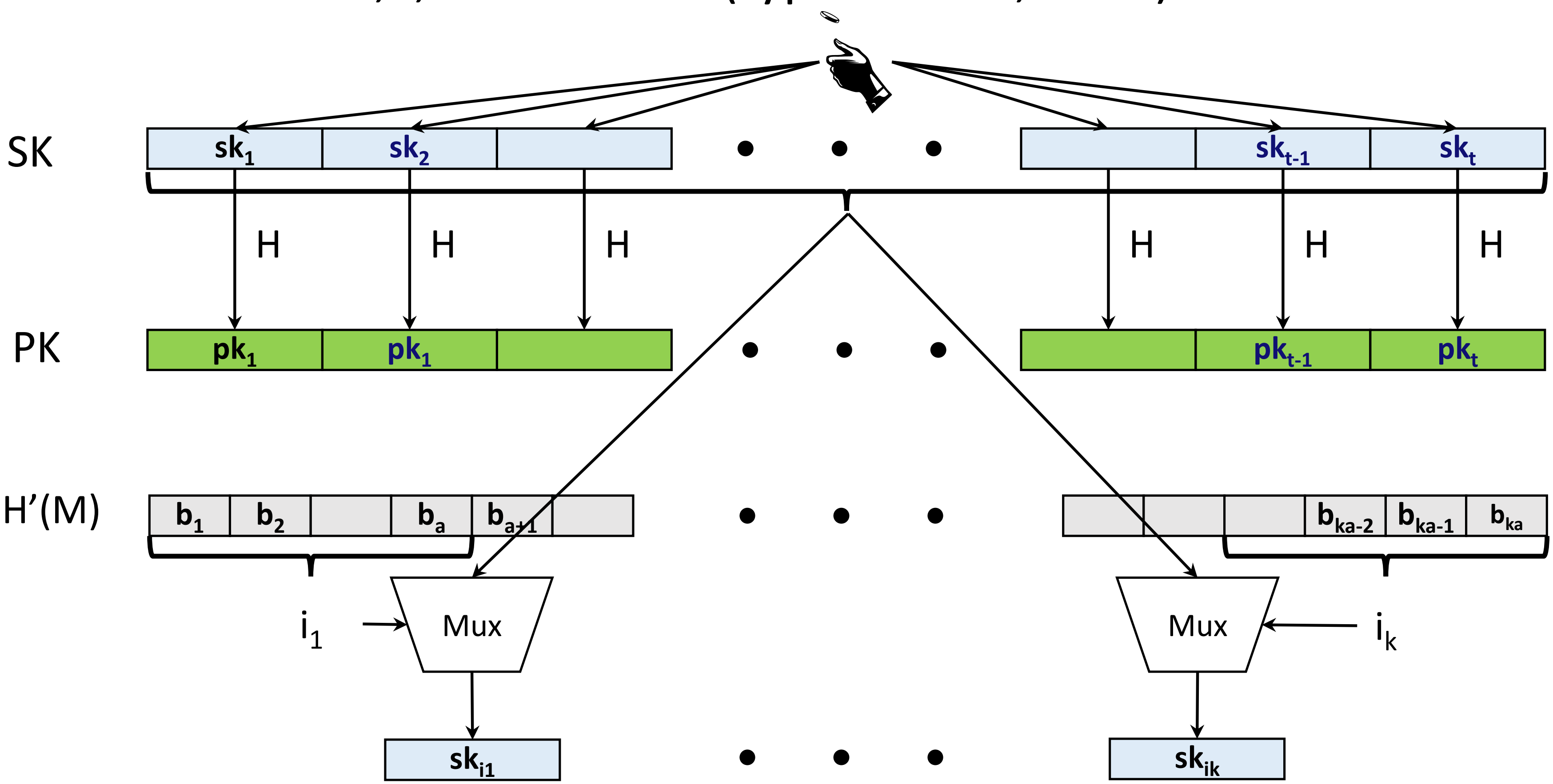
Parameters $t=2^a$,k, with m = ka (typical a=16, k=32)



HORS

Message M, OWF H, CRHF H'

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: $\mathrm{Succ}_{\text{r-SSR}}(A, q) = q\left(\frac{rk}{t}\right)^k$

$\longrightarrow$ Security shrinks with each signature!

HORST

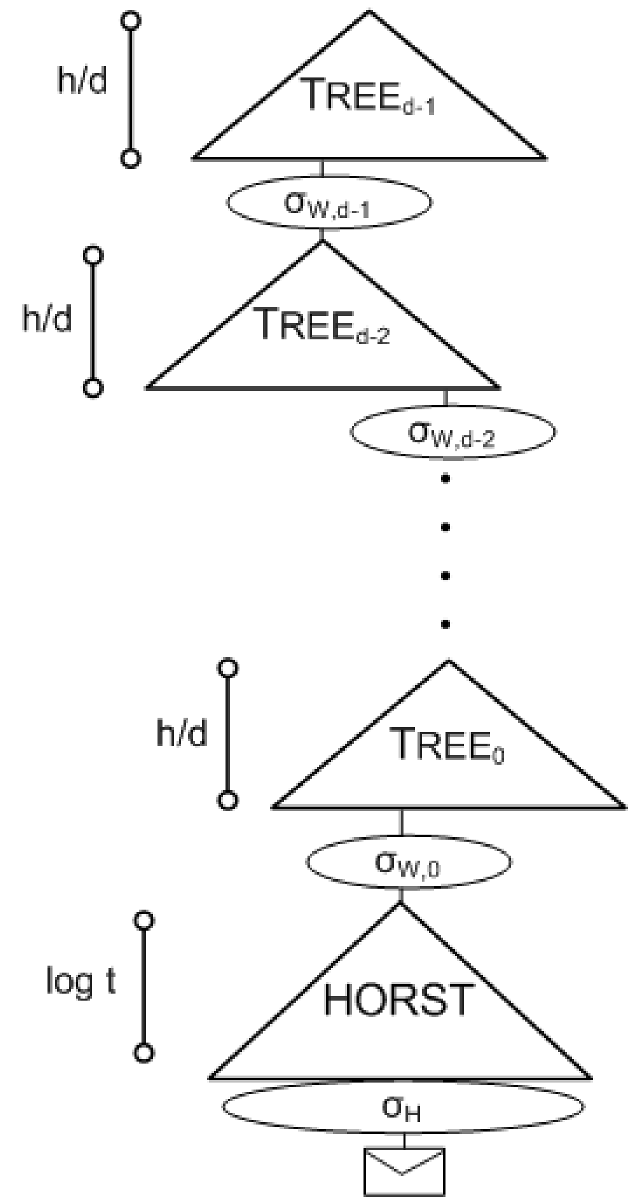
Using HORS with MSS requires adding PK (tn) to MSS signature.

HORST: Merkle Tree on top of HORS-PK

- New PK = Root
- Publish Authentication Paths for HORS signature values
- PK can be computed from Sig
- With optimizations: $tn \longrightarrow (k(\log t - x + 1) + 2^x)n$
  - E.g. SPHINCS-256: 2 MB $\rightarrow$ 16 KB
- Use randomized message hash

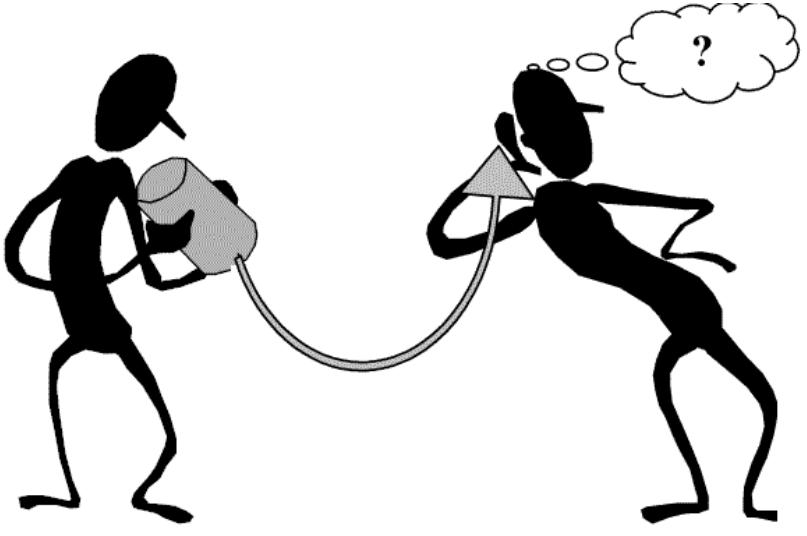
SPHINCS

- 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



Thank you!

Questions?



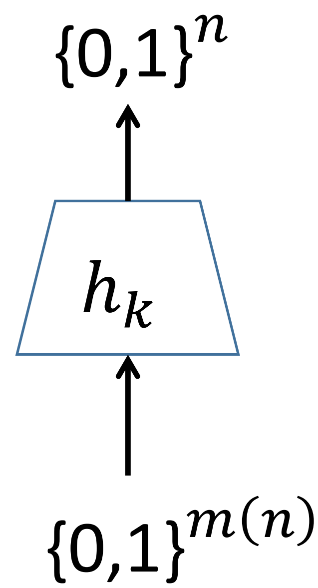
For references & further literature see
https://huelsing.wordpress.com/hash-based-signature-schemes/literature/

(Hash) function families

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

- $m(n) \geq n$

- „efficient“



One-wayness

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

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

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

$$y_c \leftarrow h_k(x)$$

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

y_c, k

x^*

Collision resistance

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

$h_k \overset{\$}{\leftarrow} H_n$

Success if
$h_k(x_1^*) = h_k(x_2^*)$

Second-preimage resistance

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

$h_k \xleftarrow{\$} H_n$

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

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

Undetectability

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

$h_k \overset{\$}{\leftarrow} H_n$

$b \overset{\$}{\leftarrow} \{0,1\}$

If $b = 1$

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

$\quad y_c \leftarrow h_k(x)$

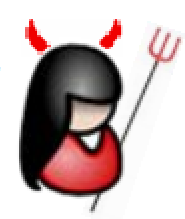
else

$\quad y_c \overset{\$}{\leftarrow} \{0,1\}^n$

y_c, k

$\downarrow$

$\mathcal{A}$



$\downarrow$

b^*

Pseudorandomness

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