SPHINCS: practical stateless hash-based signatures

D. J. Bernstein, D. Hopwood, A. Hülsing,
T. Lange, R. Niederhagen, L. Papachristodoulou,
P. Schwabe, and Z. Wilcox O'Hearn

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Digital Signatures are Important!

Software updates

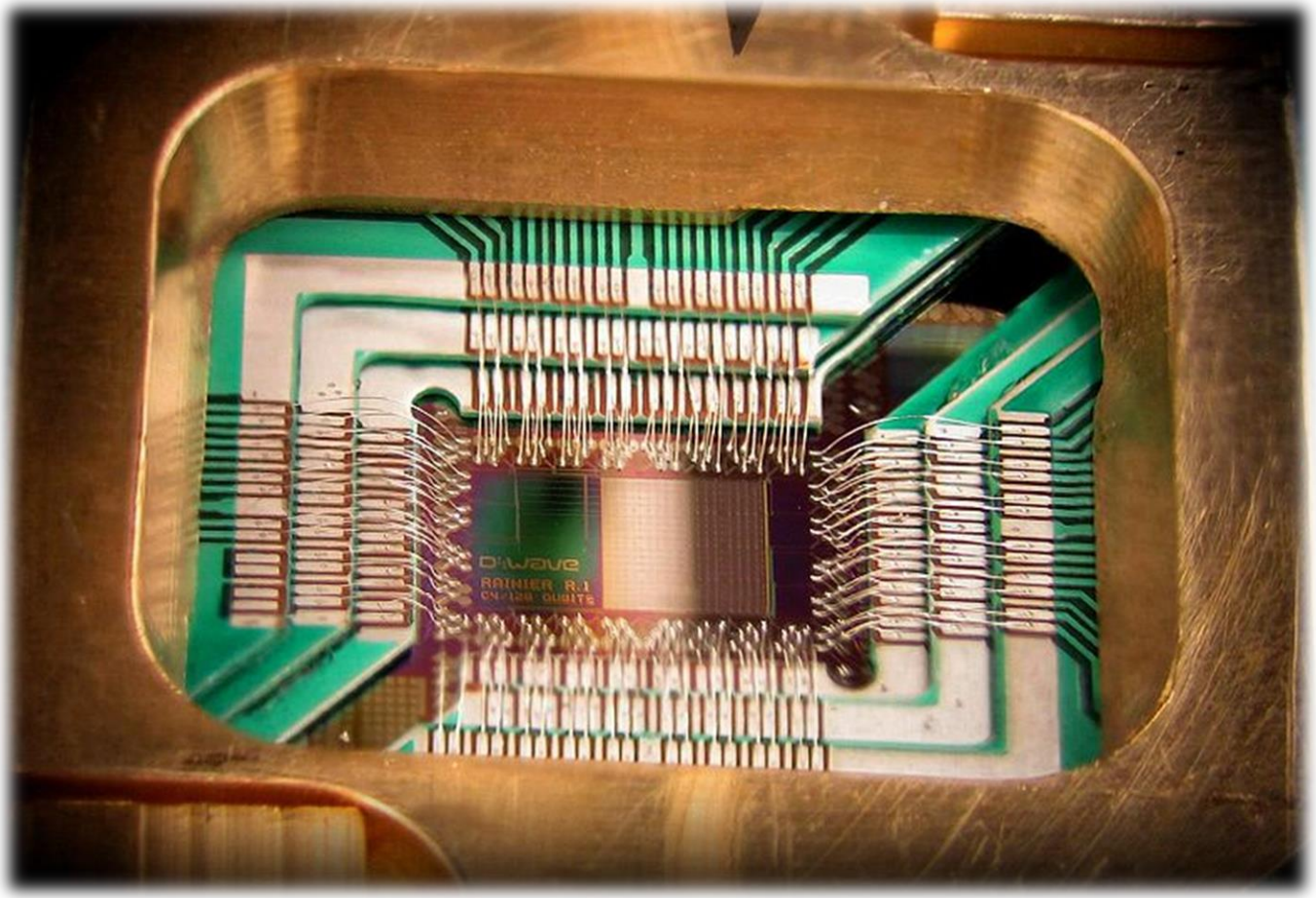
amazon

E-Commerce

ebay™

… and many others

What if…



Problem?

IBM 2012: ***„…optimism about superconducting qubits and the possibilities for a future quantum computer are***

„Want to factor a 2000-bit number?

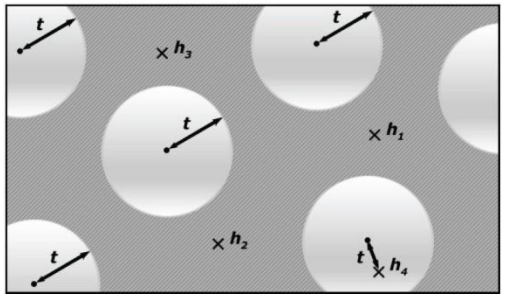
- ***Quantum computer runtime = 24h***
- ***Number of physical qubits required = 500M***
- ***Power required ~ dedicated nuclear power plant***
- ***Lead time ~5 (research) + 10 (development) years***
- ***Dollar required ~1 billion (CERN ~ 10 billion)“***

Matteo Mariantoni, PQ-Crypto 2014

Post-Quantum Signatures

Lattice, MQ, Coding

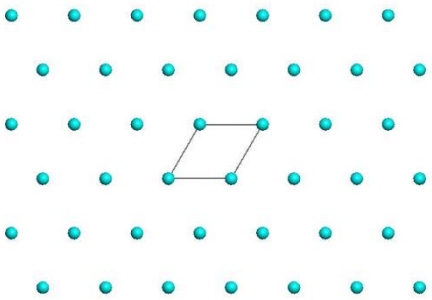
- Signature and/or key sizes
- Runtimes
- Secure parameters



$$y_1 = x_1^2 + x_1x_2 + x_1x_4 + x_3$$

$$y_2 = x_3^2 + x_2x_3 + x_2x_4 + x_1 + 1$$

$$y_3 = \ldots$$



Hash-based Signature Schemes [Mer89]

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

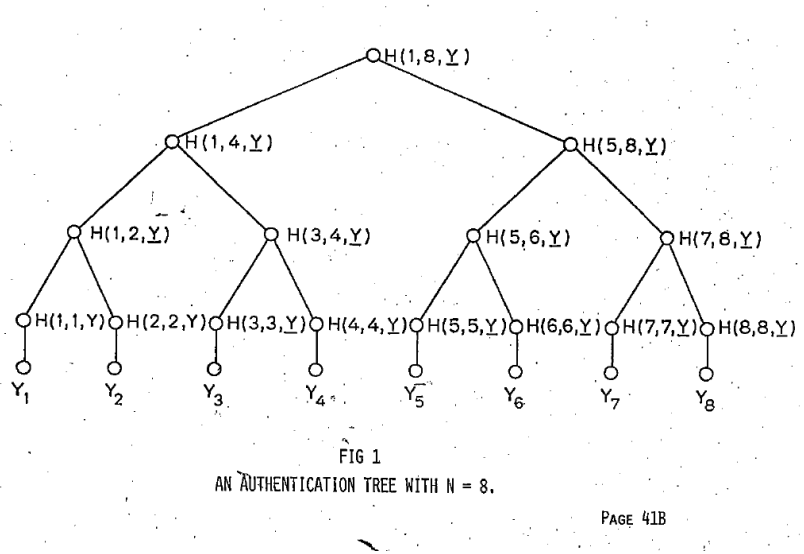
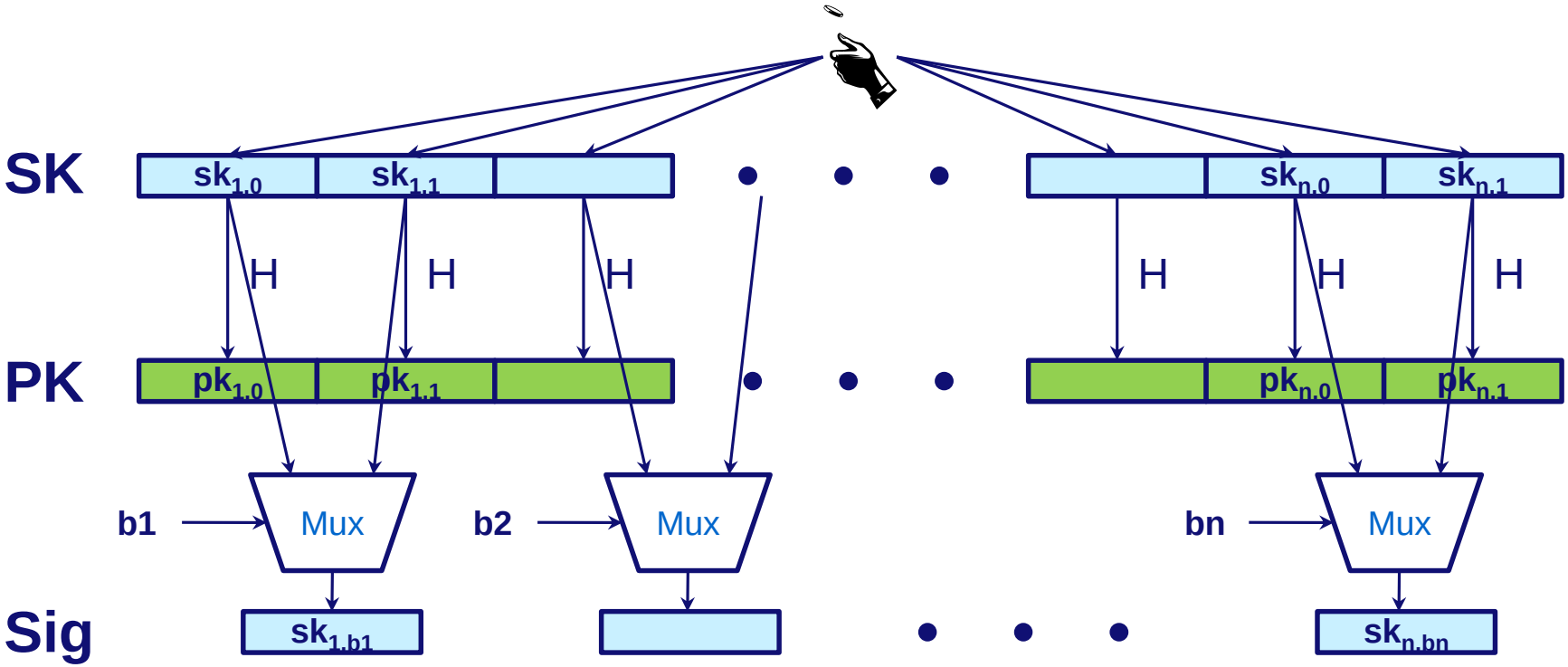


FIG 1
AN AUTHENTICATION TREE WITH N = 8.

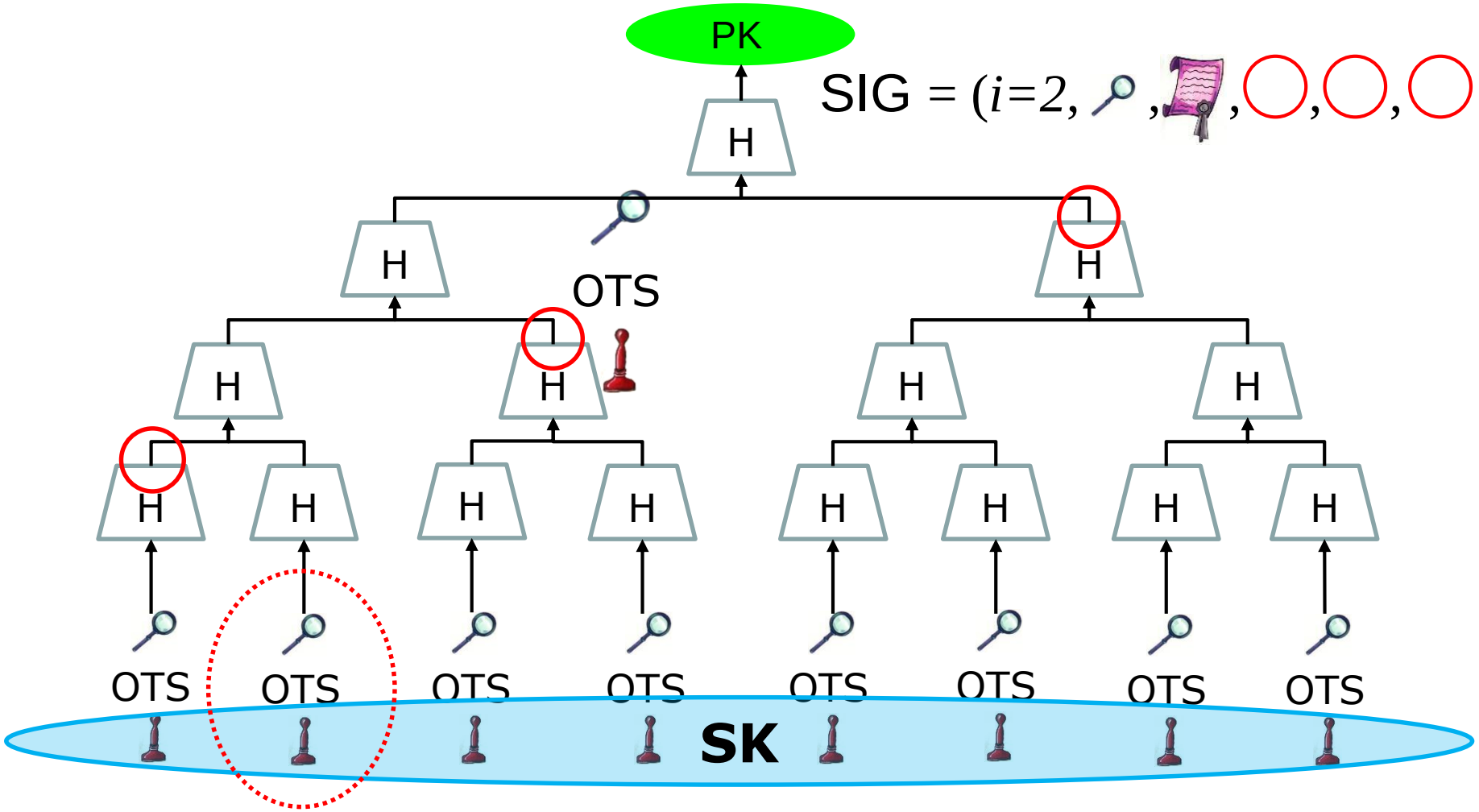
PAGE 41B

Lamport-Diffie OTS [Lam79]

Message M = b1,…,bn, OWF H ***** **= n bit**



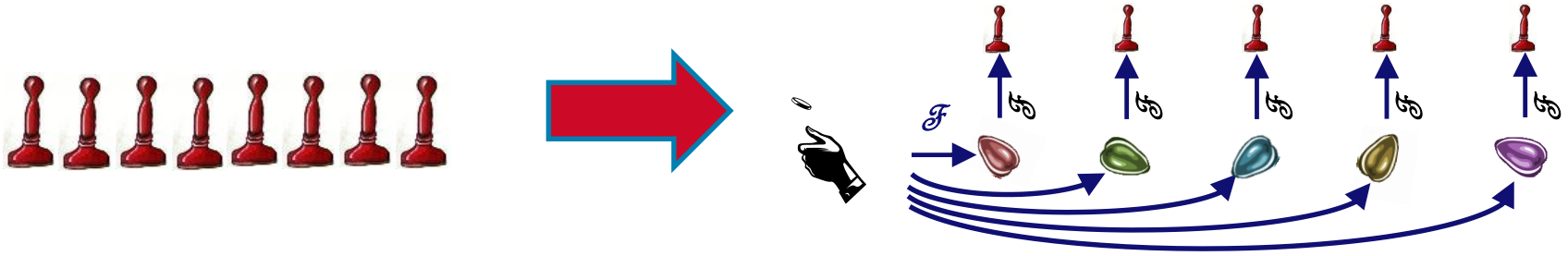
Merkle's Hash-based Signatures



Known Improvements [BGD+06]

- Pseudorandom Key Generation:

  SK $2^h n \to n$

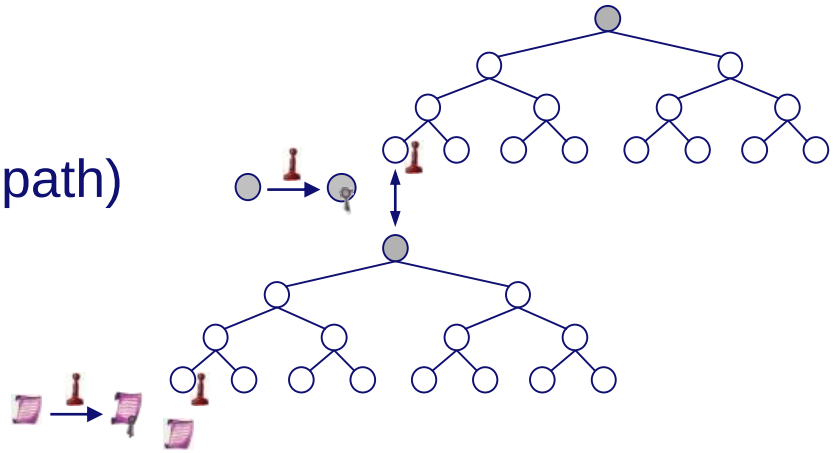


- Hypertree:

  Key generation

  (== Building Trees on one path)

  $\mathcal{O}(2^h) \to \mathcal{O}(d2^{h/d})$



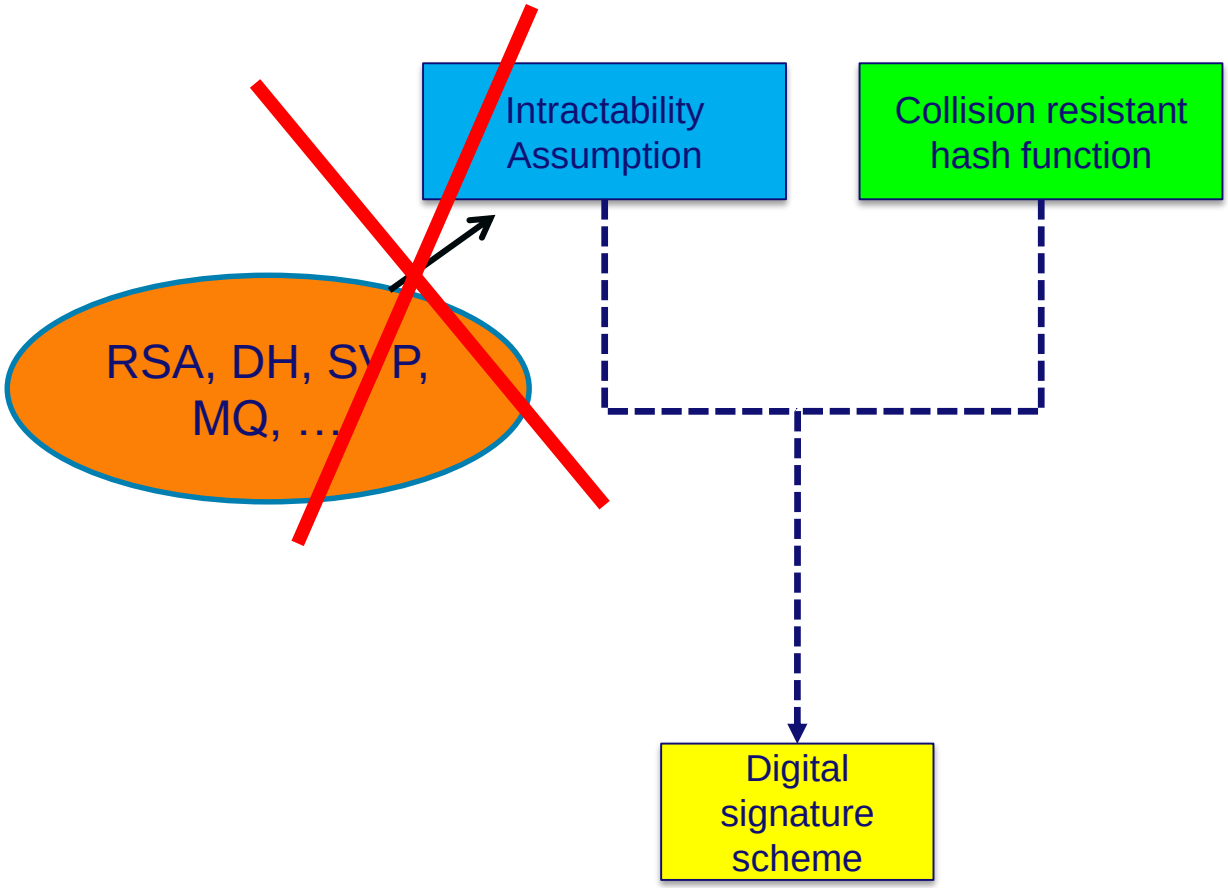
Known Improvements, cont'd

- **Winternitz OTS [Mer'89, …, BDE+11,Hül13]**
  - **Smaller Sigs**
  - **Minimum Security Assumptions**

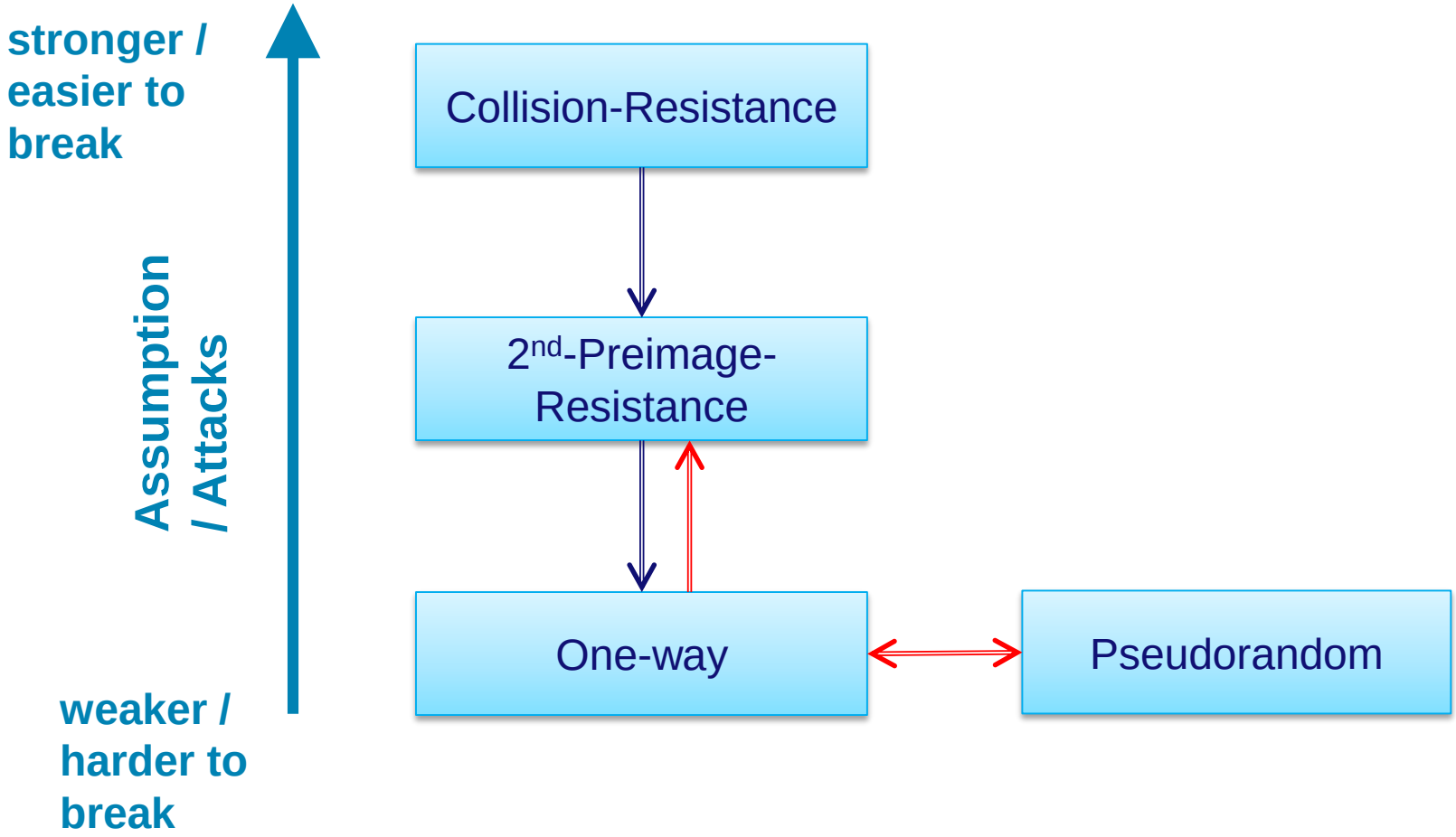
- **Minimum Security Assumptions / Collision-Resilient Scheme [BDH11]**
  - **XMSS requires only PRF & SPR**

Advantages of Hash-based Signatures

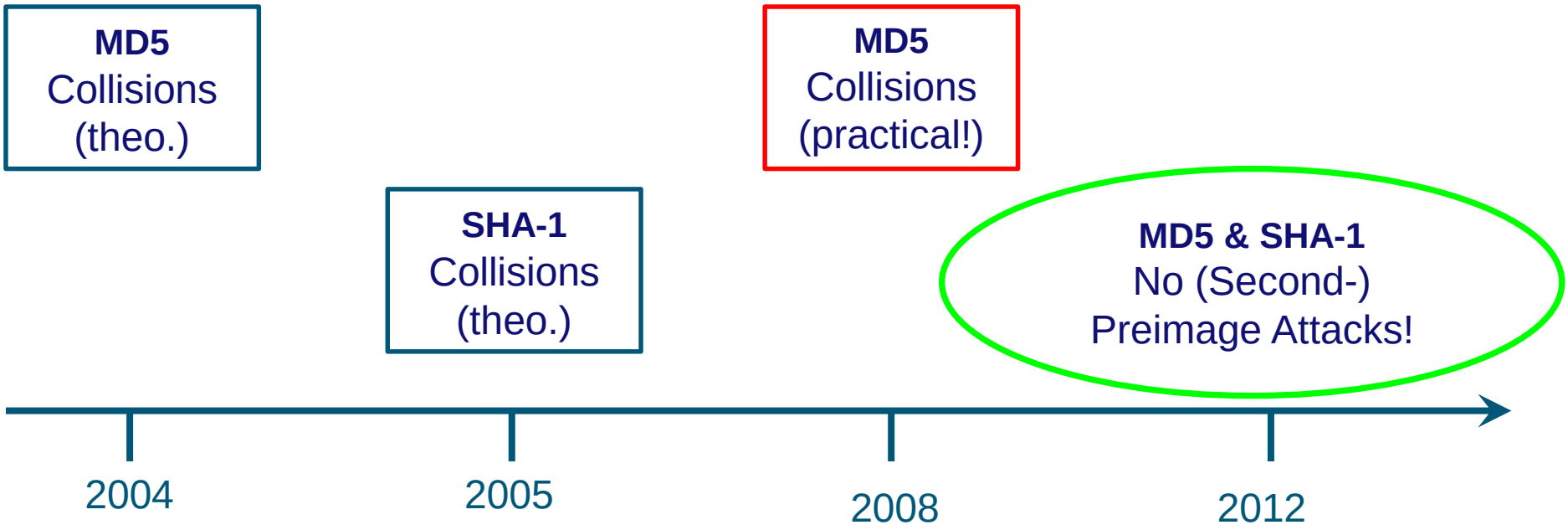
RSA – DSA – EC-DSA...



Early Warning System
Hash-function properties



Early Warning System
Attacks on Hash Functions



Easy Redundancy

Hash-Combiner

- Collision-Resistance / 2nd-Preimage-Resistance:

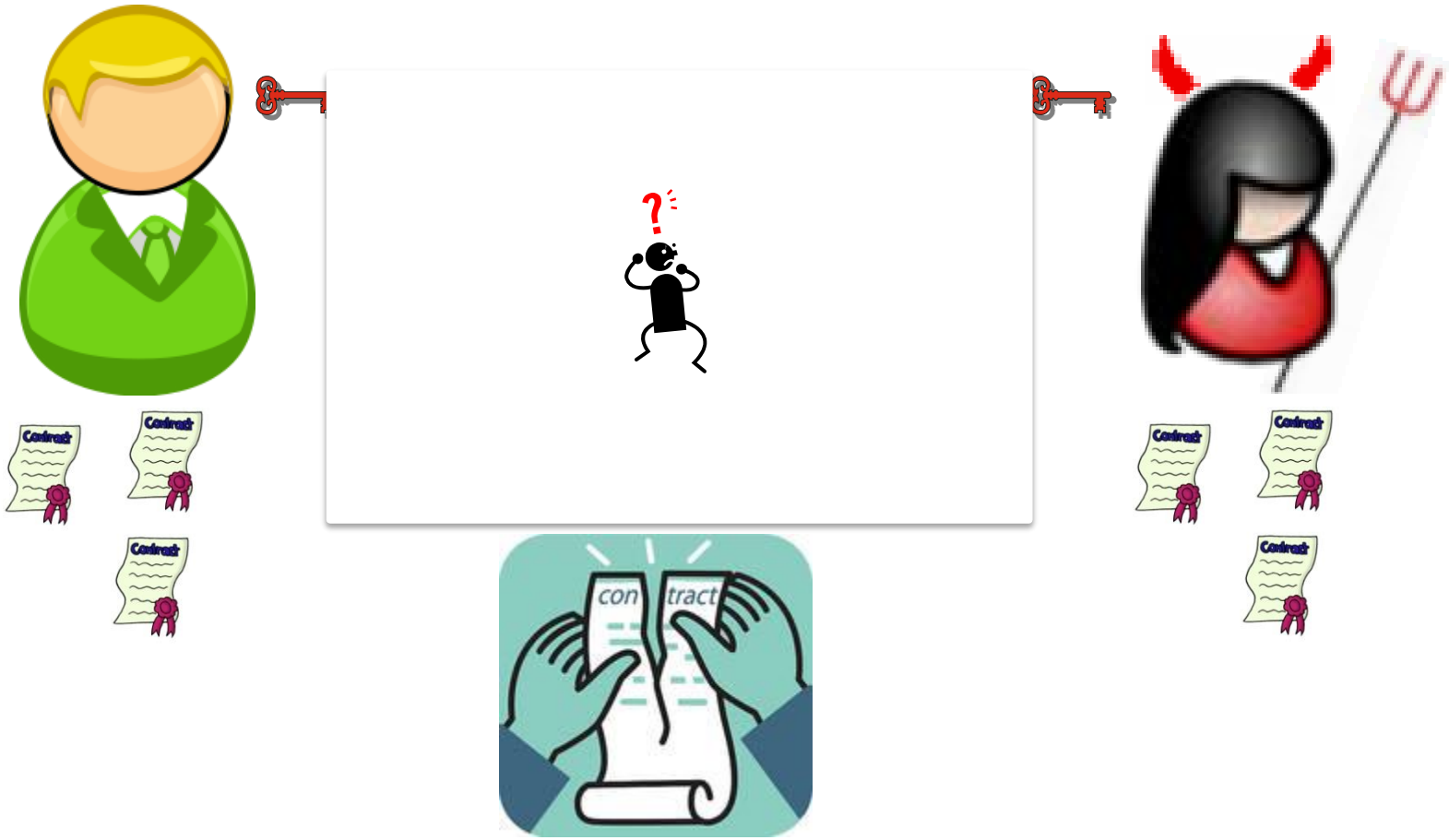
$$h_k(x) = g_k(x) \| f_k(x)$$

- PRF:

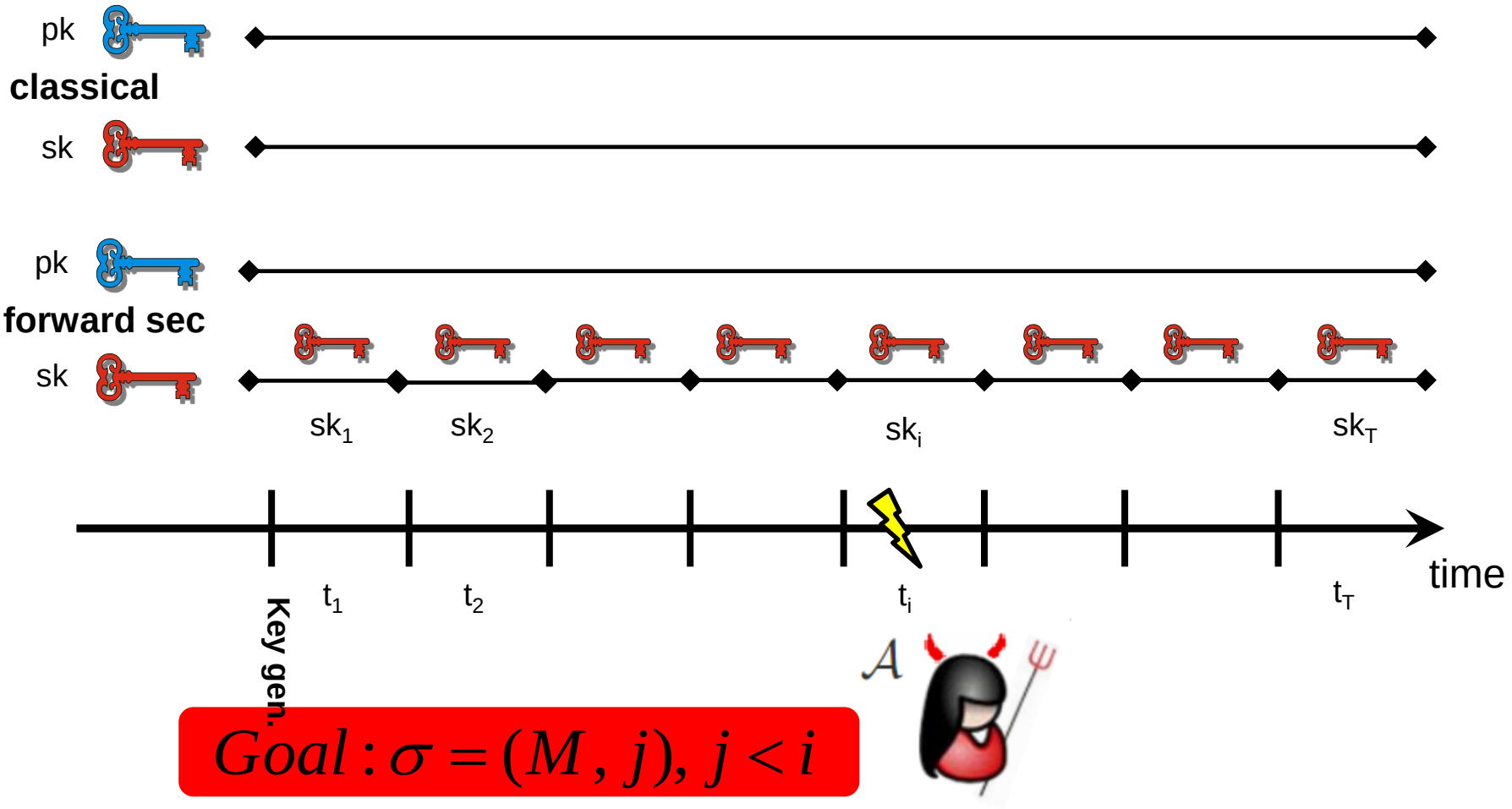
$$h_k(x) = g_k(x) \oplus f_k(x)$$

- **No sudden break**
- **Replaces double signature**
- **Signature size only grows by h*n**
- **Runtime ~ doubled**

Forward Security



Forward Security - cont'd



$$Goal : \sigma = (M, j), j < i$$

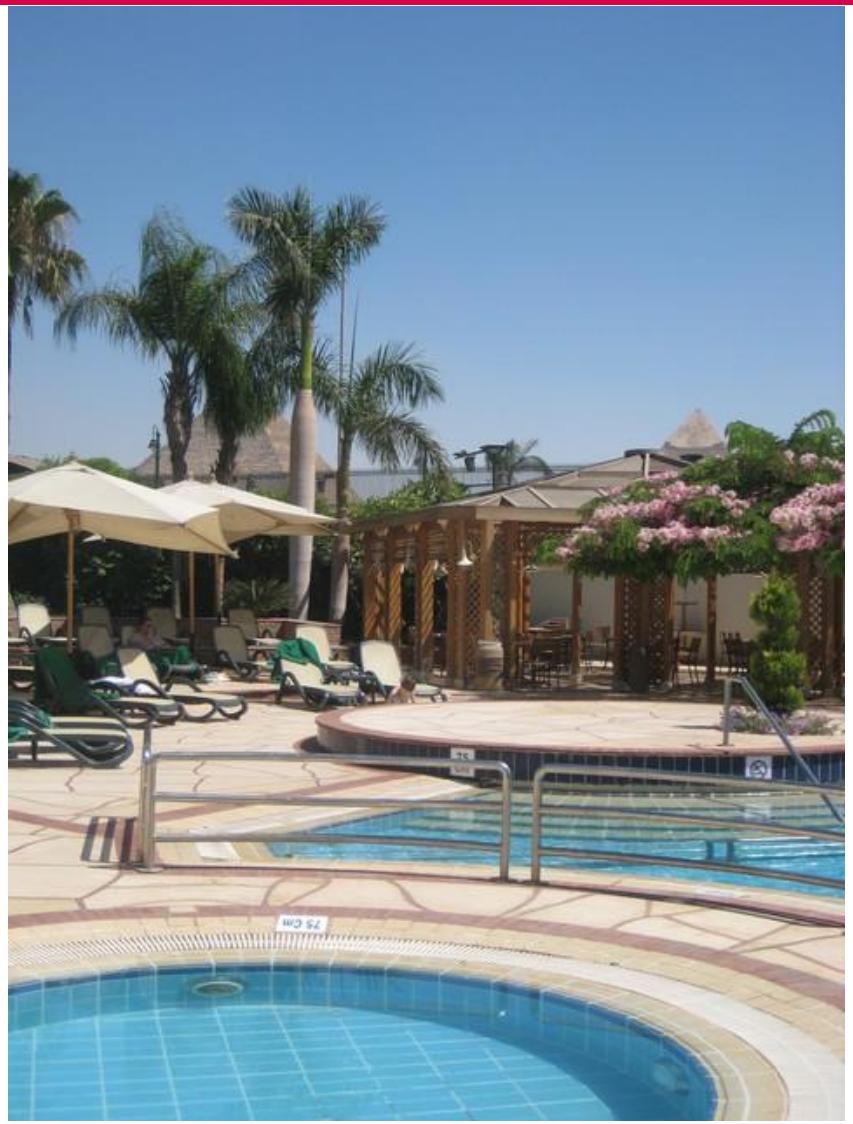
Long-Standing Problem: Statefulness

- No problem in many cases.
  - Qualified signatures,
  - Keys on smartcard, ...
- Necessary for forward-security!

But:

- Key back-ups undermine security
- Parallel use of key problematic
  - Multi-threading scenarios,
  - Load balancing...
- Do not fit standard API

SPHINCS: Stateless Practical Hash-based Incredibly Nice Cryptographic Signatures



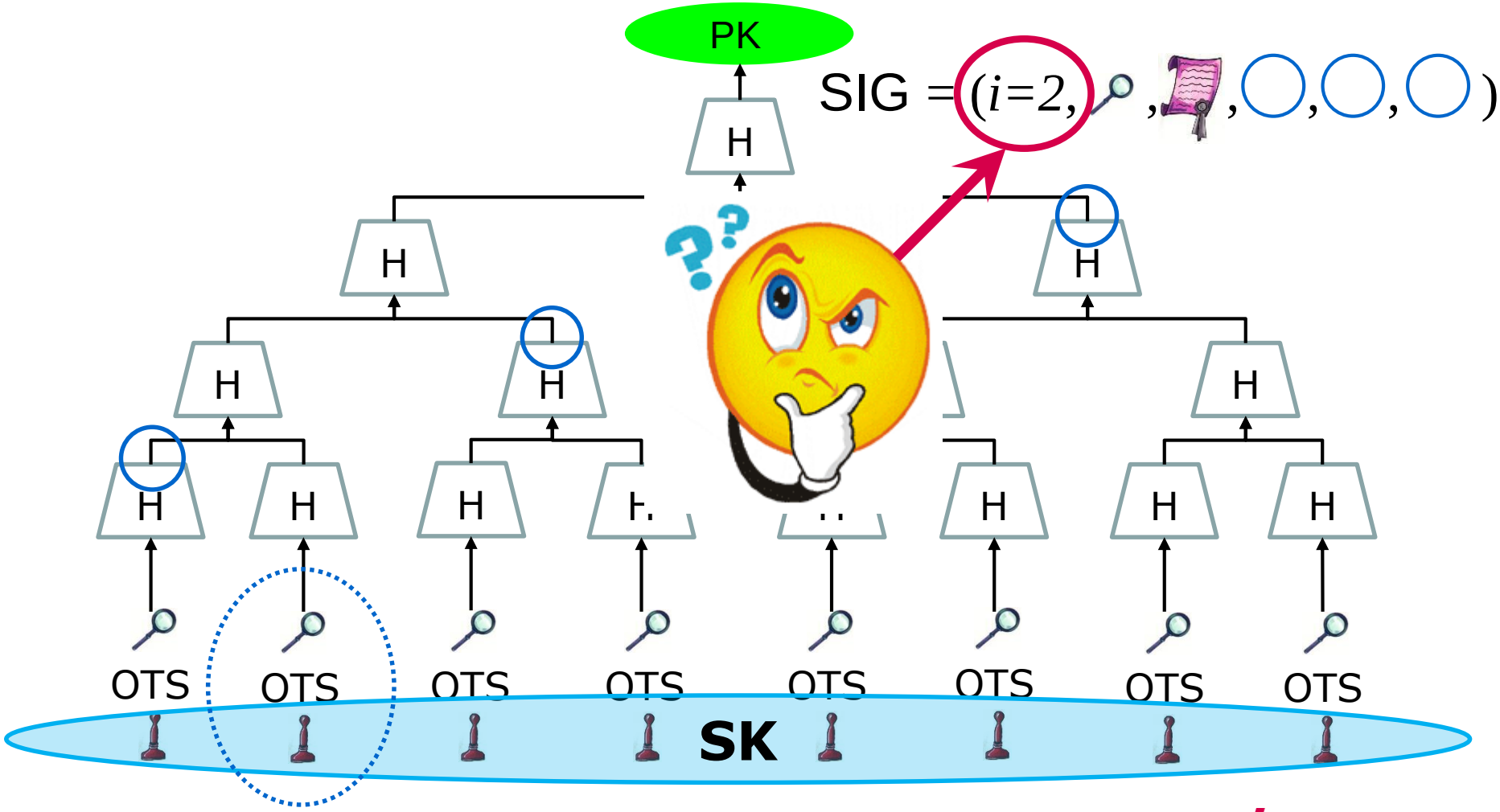
Goals

- **Stateless**
- **128bit Quantum Security**
- **Practical Speed**
- **Practical Signature Size**

How to Eliminate the State

Straight Forward

- **Run MSS without State**



Approach 1: Message Hash

i = Hash(Message);

- 128bit Quantum Sec.

  $\rightarrow n = 256$ bit Hash [Ber09]

  $\rightarrow$ #Indices = 2^{256}

  $\rightarrow h = n = 256$

- h depends on n!
- Best we can do:

  $t_{Sign} \approx n^3 / \log n \; t_{Hash} = 2M \; t_{Hash} \approx 15$ min*

  $|Sig| \approx n^3 / \log n > 256$ kb

* (OpenSSL SHA2)

Approach 2: Random Index

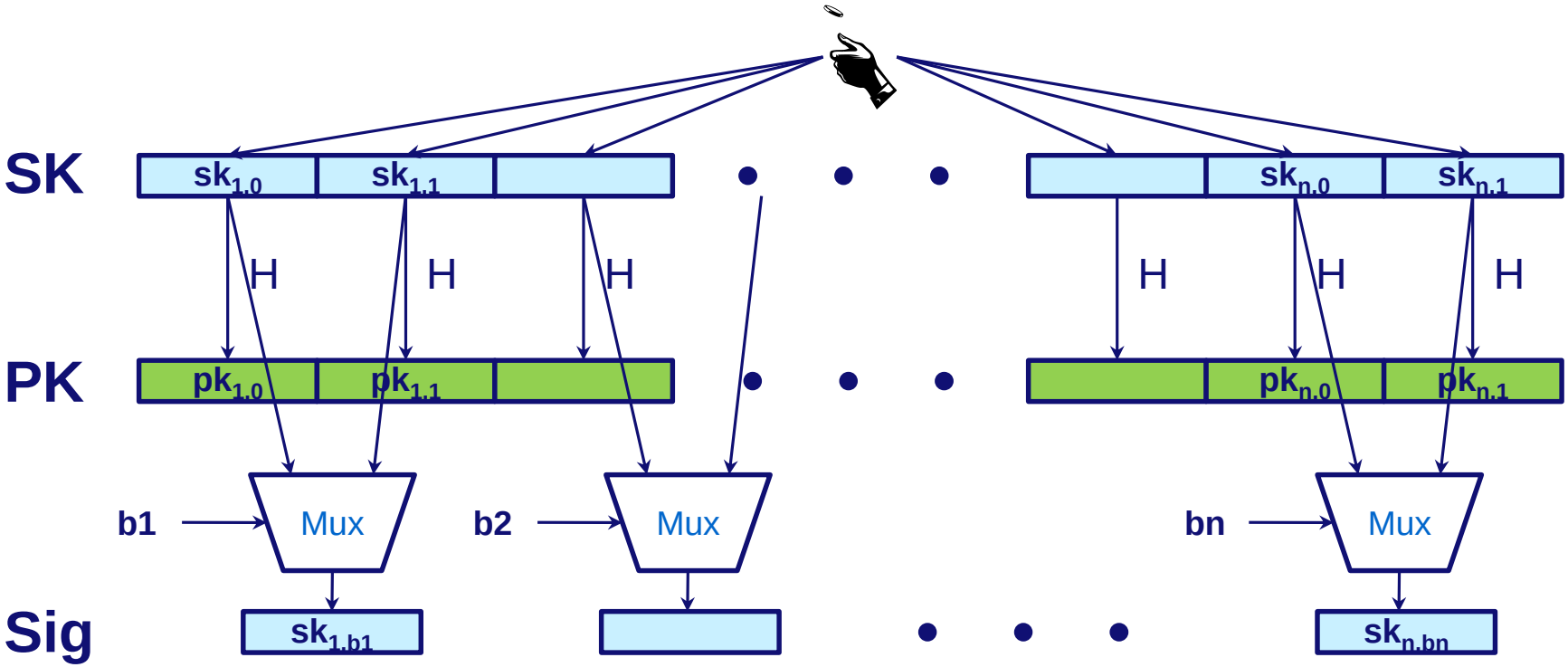
$$I \leftarrow_{\$} U_{\#Indices}$$

- 128bit Quantum Sec.
  - → Sampled by Signer
    - → #Indices determined by collision prob.
    - → #Indices = 2^{256}
    - → $h = 256$
- Impossible to make this efficient, again… BUT
  - h independent of n
  - Statistical collision probability NOT collision resistance

Few-Time Signature Schemes

Recap LD-OTS

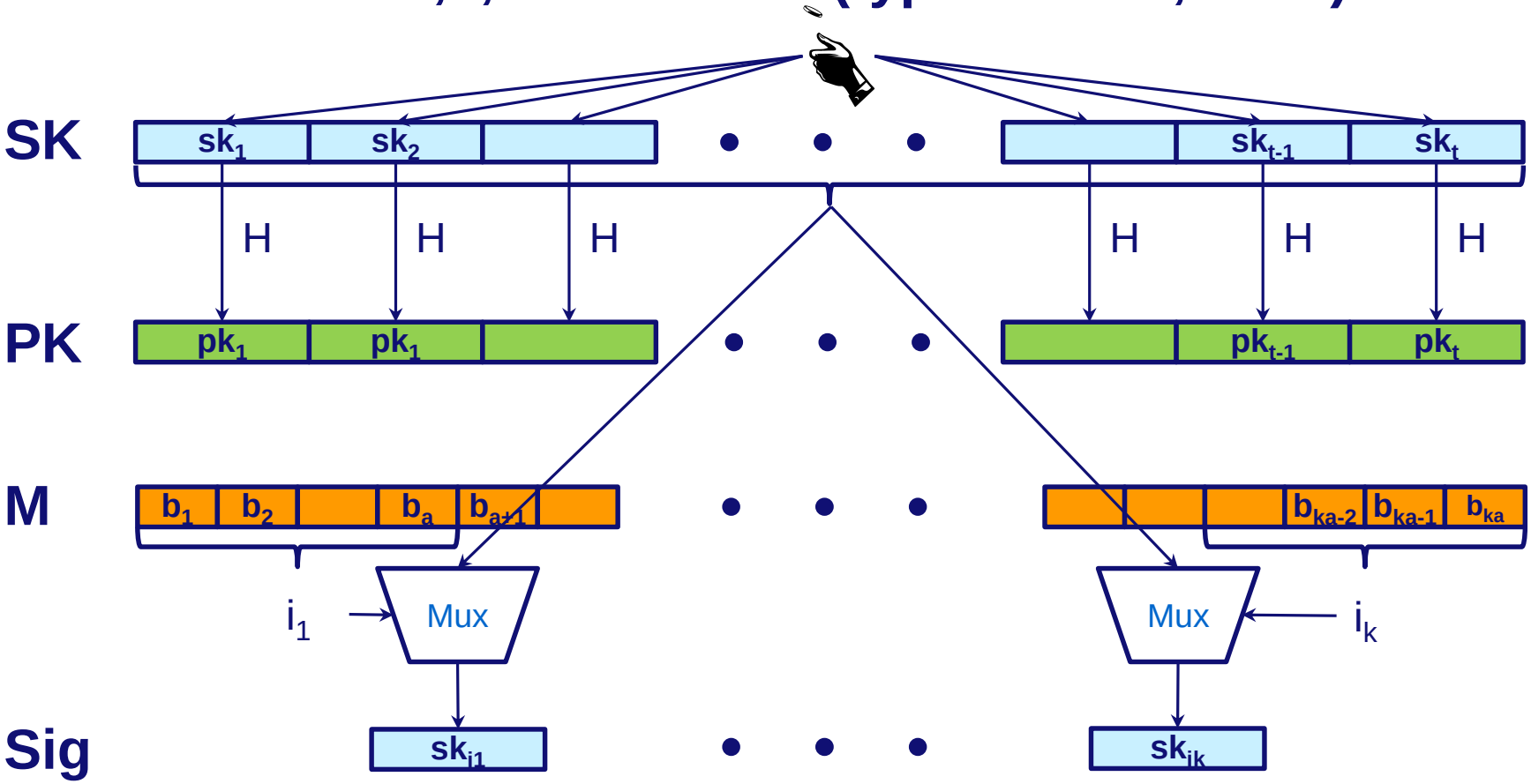
Message M = b1,…,bn, OWF H ***** **= n bit**



HORS [RR02]

Message M = $b_1,\ldots,b_m$, OWF H [*] **= n bit**

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



HORS Security

- Message M = Hash(msg)
- M mapped to k element index set $M^i \in \{1,..,t\}^k$
- Each signature publishes k out of t secrets
- Either break one-wayness or…

- r-Subset-Resilience: After seeing index sets M^i_j for r messages msg_j, $1 <= j <= r$, hard to find $msg_{r+1} \neq msg_j$ such that $M^i_{r+1} \in \cup_{1<=j<=r} M^i_j$.

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

→ Security shrinks with each signature!

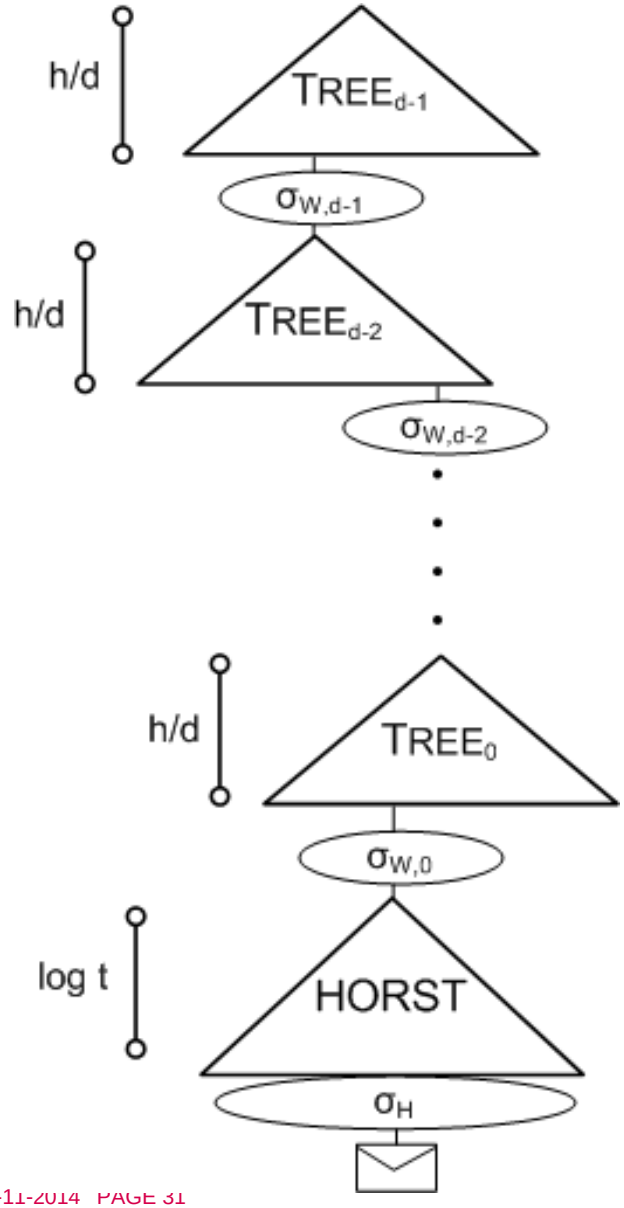
HORST

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

- **HORST: Build a Merkle Tree on top of HORS-PK**
  - **PK = Root**
  - **Publish Authentication 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**

SPHINCS

SPHINCS Signature



SPHINCS Key Ideas

- Use HORST key pairs to sign messages
- Authenticate HORST key pairs using hypertree (of MSS trees)
- Use random index

- Select Parameters such that

$$\text{sum}_{r \in [0,\infty)}(\Pr[\text{r-times index collision}] * \text{Succ}_{\text{r-SSR}}(A)) = \text{negl}(n)$$

SPHINCS-256

Table 1: SPHINCS-256 parameters and functions for the 128-bit post-quantum security level and resulting signature and key sizes.

Parameter	Value	Meaning
n	256	bitlength of hashes in HORST and WOTS
m	512	bitlength of the message hash
h	60	height of the hyper tree
d	12	layers of the hyper tree
w	16	Winternitz parameter used for WOTS signatures
t	2^{16}	number of secret-key elements of HORST
k	32	number of revealed secret-key elements per HORST signature
Functions		
Hash $\mathcal{H}$:	$\mathcal{H}(R, M) = \text{BLAKE-512}(R\|\|M)$	
PRF $\mathcal{F}_a$:	$\mathcal{F}_a(A, K) = \text{BLAKE-256}(K\|\|A)$	
PRF $\mathcal{F}$:	$\mathcal{F}(M, K) = \text{BLAKE-512}(K\|\|M)$	
PRG G_λ:	$\text{G}_\lambda(\text{SEED}) = \text{ChaCha12}_{\text{SEED}}(0)_{0,\ldots,\lambda-1}$	
Hash F:	$\text{F}(M_1) = \text{CHOP}(\pi_{ChaCha}(M_1\|\|C), 256)$	
Hash H:	$\text{H}(M_1\|\|M_2) = \text{CHOP}\left(\pi_{ChaCha}\left(\pi_{ChaCha}\left(M_1\|\|C\right) \oplus \left(M_2\|\|0^{256}\right)\right), 256\right)$	
Sizes		
Signature size:	41000 bytes	
Public-key size:	1056 bytes	
Private-key size:	1088 bytes	

SPHINCS-256 Speed

- **Key generation:** 3,051,562 cycles
- **Verification:** 1,369,060 cycles
- **Signature:** 47,466,005 cycles

- **Still hundreds of messages per second on a modern 4-core 3.5GHz Intel CPU (13ms / Sig)**

- **Remember: Straight Forward**

$$t_{Sign} \approx 15 \text{ min}^*$$
$$|Sig| > 256 \text{ kb}$$

In Paper (http://eprint.iacr.org/2014/795)

- Standard model security reduction without collision resistance
- Complexity of generic quantum attacks
- Efficient fixed-input length hashing
- Optimized implementation

Conclusion

- **If you can live with a state: Go for XMSS.**

- **Otherwise:**
  - **Go for Sphincs-256!**
  - **First stateless signature scheme with post-quantum secure parameters**
  - **Practical Speed and Sizes**

Thank you!

Questions?