XMSS: Extended Hash-Based Signatures
(draft-huelsing-cfrg-hash-sig-xmss)

A. Hülsing, D. Butin, S.-L. Gazdag, A. Mohaisen

Hash-based Signature Schemes
[Mer89]

Only secure hash function

Security well understood

Post quantum

Fast

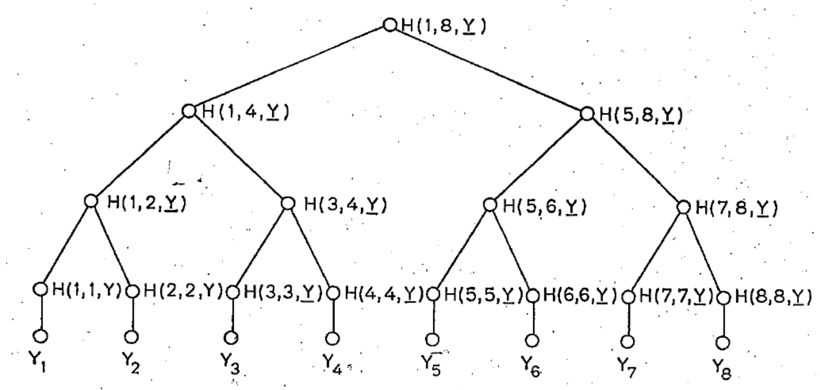
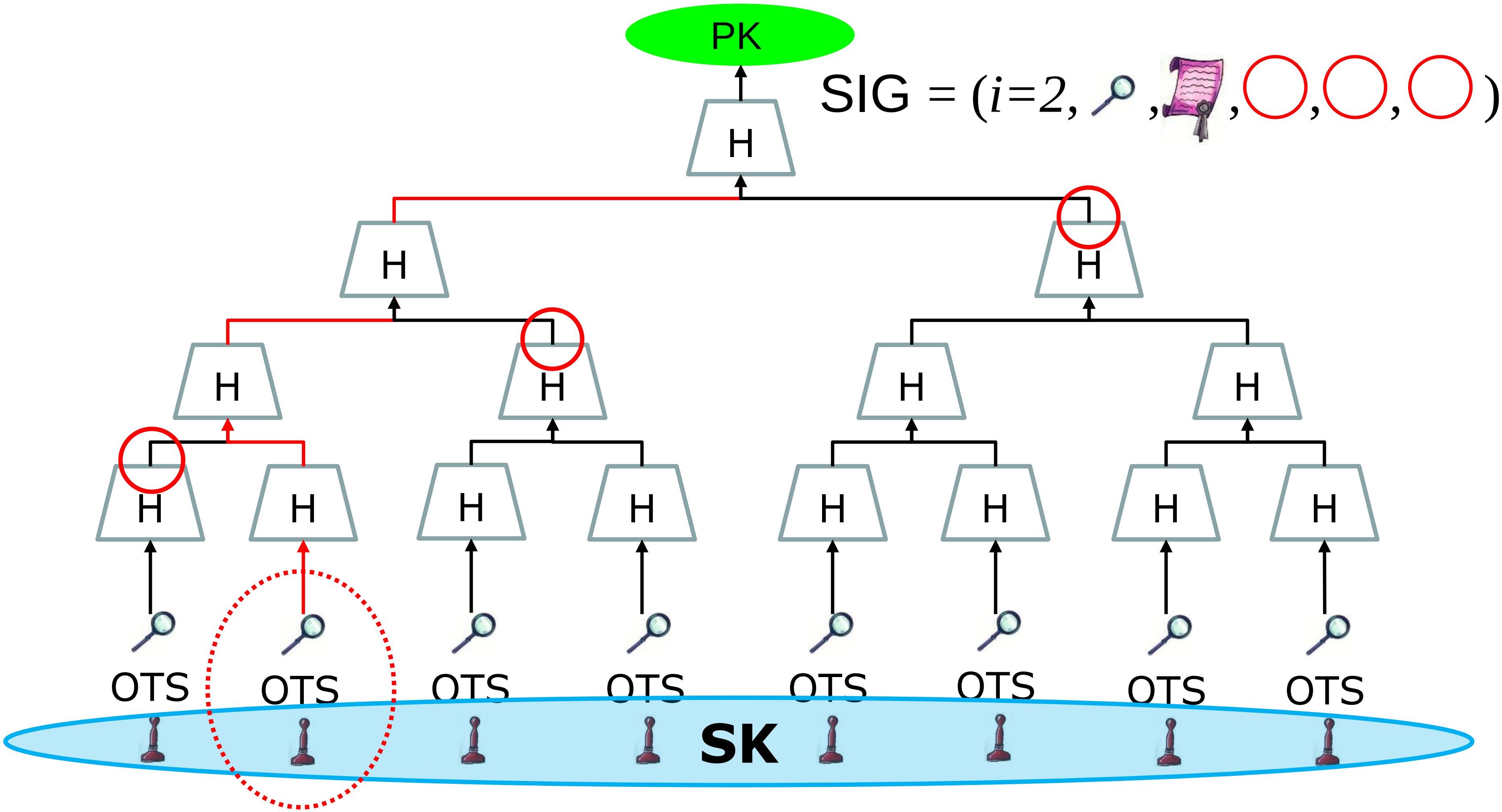


FIG 1
AN AUTHENTICATION TREE WITH N = 8.

PAGE 41B

Merkle's Hash-based Signatures



McGrew & Curcio‘2014

```
Crypto Forum Research Group                                 D. McGrew
Internet-Draft                                              M. Curcio
Intended status: Informational                          Cisco Systems
Expires: January 5, 2015                                 July 4, 2014


                        Hash-Based Signatures
                      draft-mcgrew-hash-sigs-02

Abstract

   This note describes a digital signature system based on cryptographic
   hash functions, following the seminal work in this area.  It
   specifies a one-time signature scheme based on the work of Lamport,
   Diffie, Winternitz, and Merkle (LDWM), and a general signature
   scheme, Merkle Tree Signatures (MTS).  These systems provide
   asymmetric authentication without using large integer mathematics and
   can achieve a high security level.  They are suitable for compact
   implementations, are relatively simple to implement, and naturally
   resist side-channel attacks.  Unlike most other signature systems,
   hash-based signatures would still be secure even if it proves
   feasible for an attacker to build a quantum computer.
```

Why another I-D?

- “Weaker“ assumptions on used hash function
  - -> “Stronger“ security guarantees

- Virtually unlimited number of signatures / key pair
  (Multi-Tree version)

- Smaller signatures (approx. factor 2)

- Faster key generation & signing
  (Multi-Tree version)

Schemes in the Draft

- Winternitz One Time Signature ($WOTS^+$)
- Extended Merkle (tree) signature scheme (XMSS)
- Multi-tree XMSS (XMSS^MT)

General Design Choices

Define as mandatory:

- Public key and signature format & semantics
- Verification

Leave to implementer:

- Secret key format
  - In consequence key generation
  - Many trade-offs possible
  - Does not affect interoperability
- Signature generation
  - Many trade-offs possible
  - Does not affect interoperability

Prepare for stateless hash-based signatures (future):

- SPHINCS uses XMSS^MT as subroutine

Efficient sig / pk encodings a la McGrew & Curcio

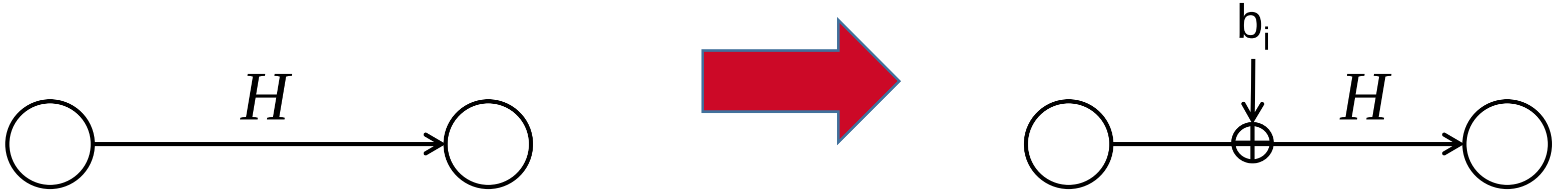
WOTS$^+$

Uses bitmasks

-> Collision-resilience

-> signature size halved

-> Tighter security reduction



Design Choices: WOTS$^+$

- Key Generation & Signing defined for random secret key
  - Works with any pseudorandom key generation method
  - We describe one pseudorandom key generation method
    - Implementers choice, does not affect interoperability
- Verification split into
  - Public key from Signature function PKfromSig
  - Comparison

  -> XMSS only uses PKfromSig

  -> Allows stand-alone usage

XMSS

Tree:

Uses bitmasks

Leafs:

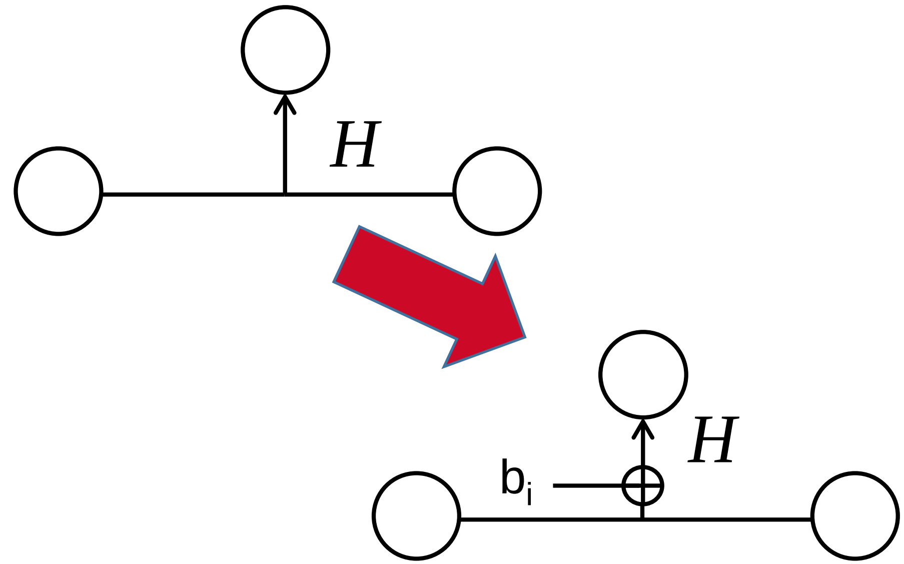
Use binary tree
with bitmasks

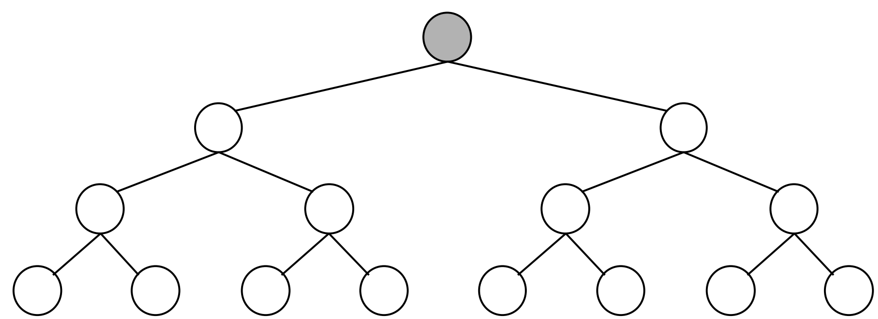
Mesage digest:

Randomized hashing

-> Collision-resilience

-> signature size halved





Design Choices: XMSS

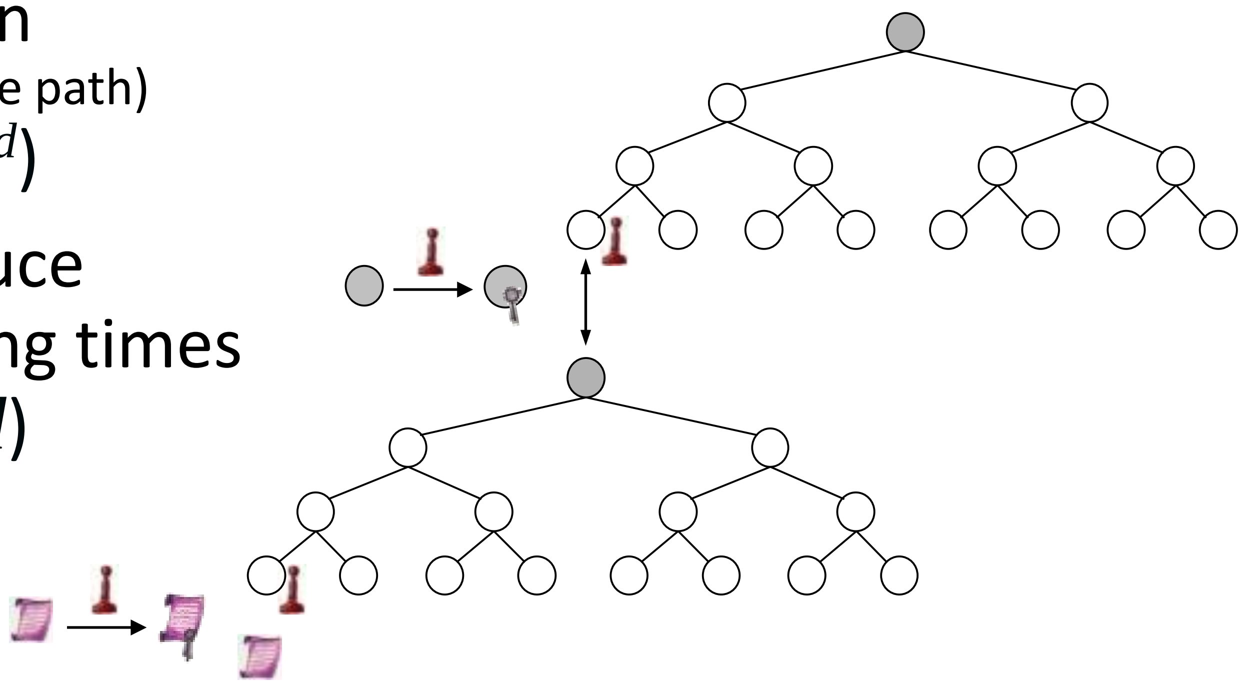
- WOTS+ instead of WOTS-PRF
- Again, describe methods as if WOTS keys stored
  - Give one pseudorandom key generation method
    - Implementers choice, does not affect interoperability
- Verification split into rootFromSig & comparison
  - To support XMSS^MT
- TreeHash for KeyGen & AuthPath computation as example

Multi-Tree XMSS

Uses multiple layers of trees

-> Key generation
(= Building Trees on one path)
$\Theta(2^h) \rightarrow \Theta(d*2^{h/d})$

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



Design Choices: Multi-tree XMSS

- Again, describe methods as if XMSS keys stored
  - Give one pseudorandom key generation method
    - Implementers choice, does not affect interoperability

- Uses XMSS Sign and Verify, w/o message hash.
- Same tree height and w for all internal trees
  -> easier implementation

Design Choices: Parameters

Parameter sets for different settings

1. Security (message digest size m, inner node size n)

	m = 256, n = 128	m = n = 256	m = n = 512
Classical Security	128 bits	256 bits	512 bits
Post-Quantum Security	64 bits	128 bits	256 bits
Internal Hash	AES-128	SHA3-256	SHA3-512
Message Digest	SHA3-256	SHA3-256	SHA3-512

Parameters, cont'd

2. WOTS$^+$:
   - w = 4, 8, 16 (optimal trade-off, easy implementation)
3. XMSS:
   - h = 10, 16, 20 (otherwise key gen too slow)
4. Multi-tree:
   - Single tree height = 5, 10, 20 (otherwise key gen too slow)
   - Total tree height h = 20, 40, 60 (> 60 unnecessary)

Parameters, cont‘d

- Many, many, many parameter sets! Too many?
- #ParameterSets
  - XMSS: 27 (+8)
  - XMSS^MT: 72 (+48)
    - will remove 18 because of statistical collision probability

  Every scenario covered?

- “Zero-Bitmasks“ parameters
  -> small PK but no collision-resilience!
  -> similar to McGrew & Curcio
  Needed?

IPR

- Based on scientific work (already published)
- No IPR claims from our side
- Not aware of others planning IPR claims

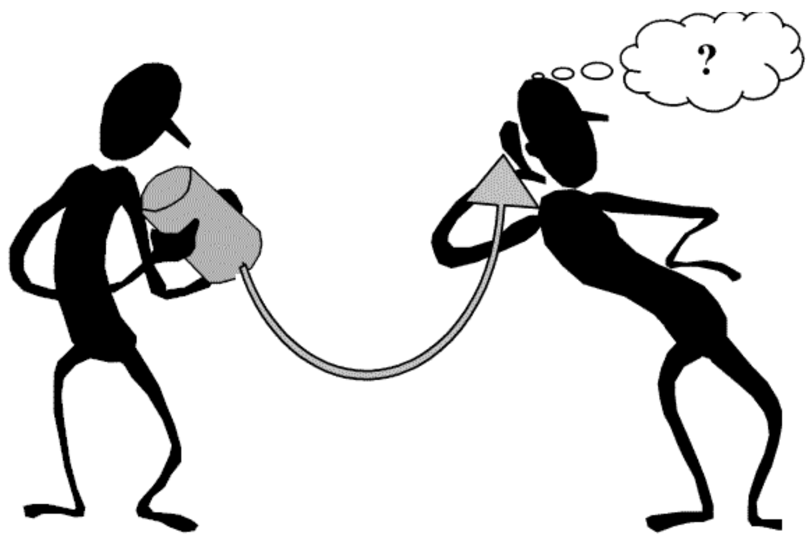
Conclusion

XMSS: New important features

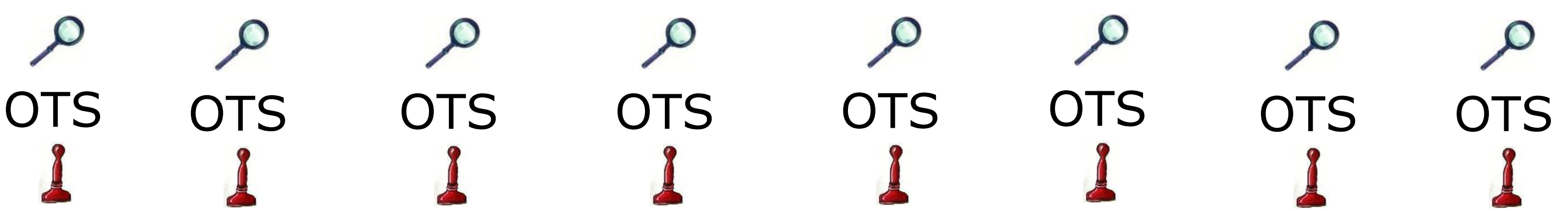
- Smaller signatures
- Faster signing & key generation
- Up to 2^{60} signatures per key pair with proposed params
- Stronger security guarantees (collision-resilience)
- Prepares for stateless schemes

Thank you!

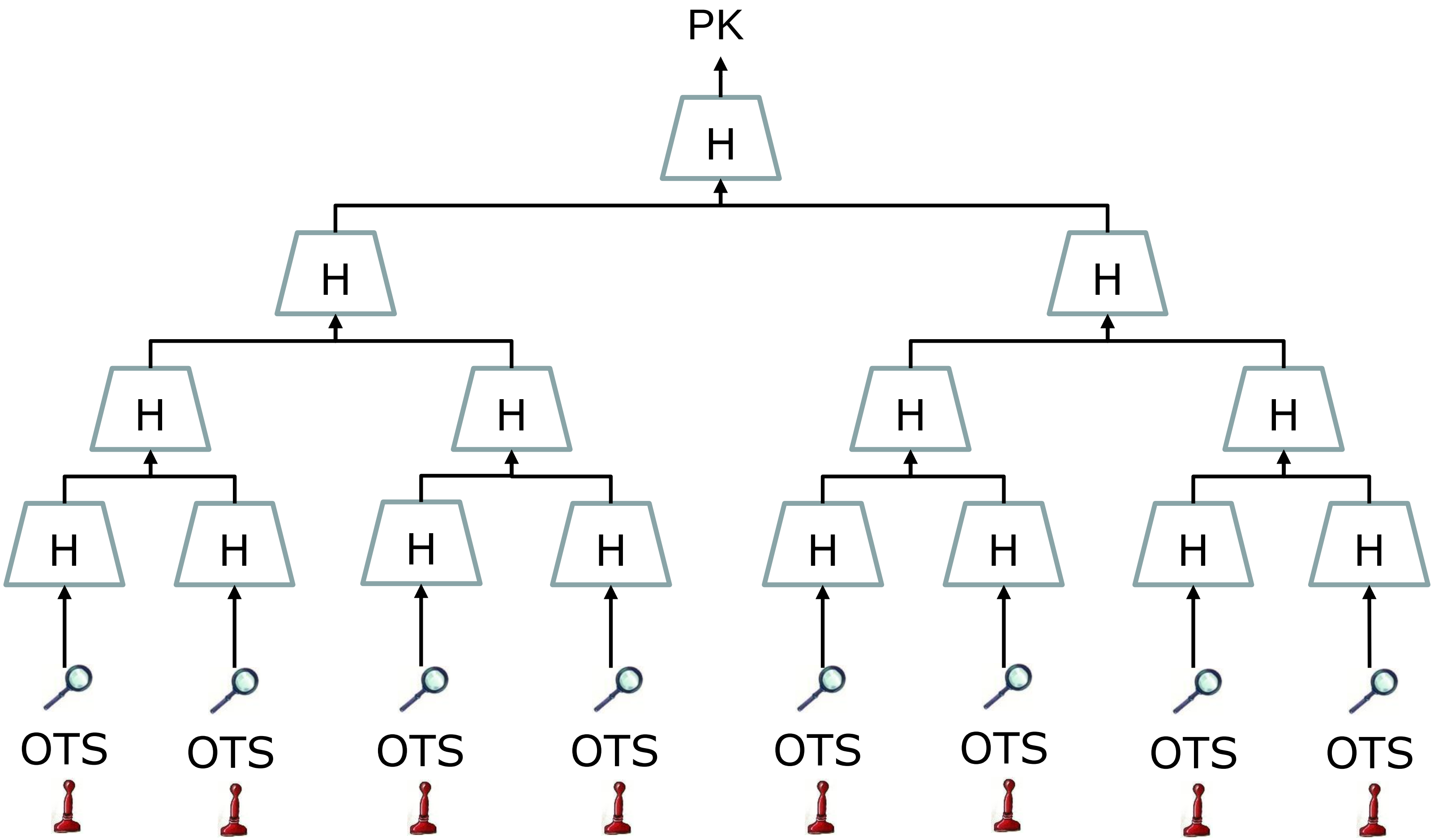
Questions?



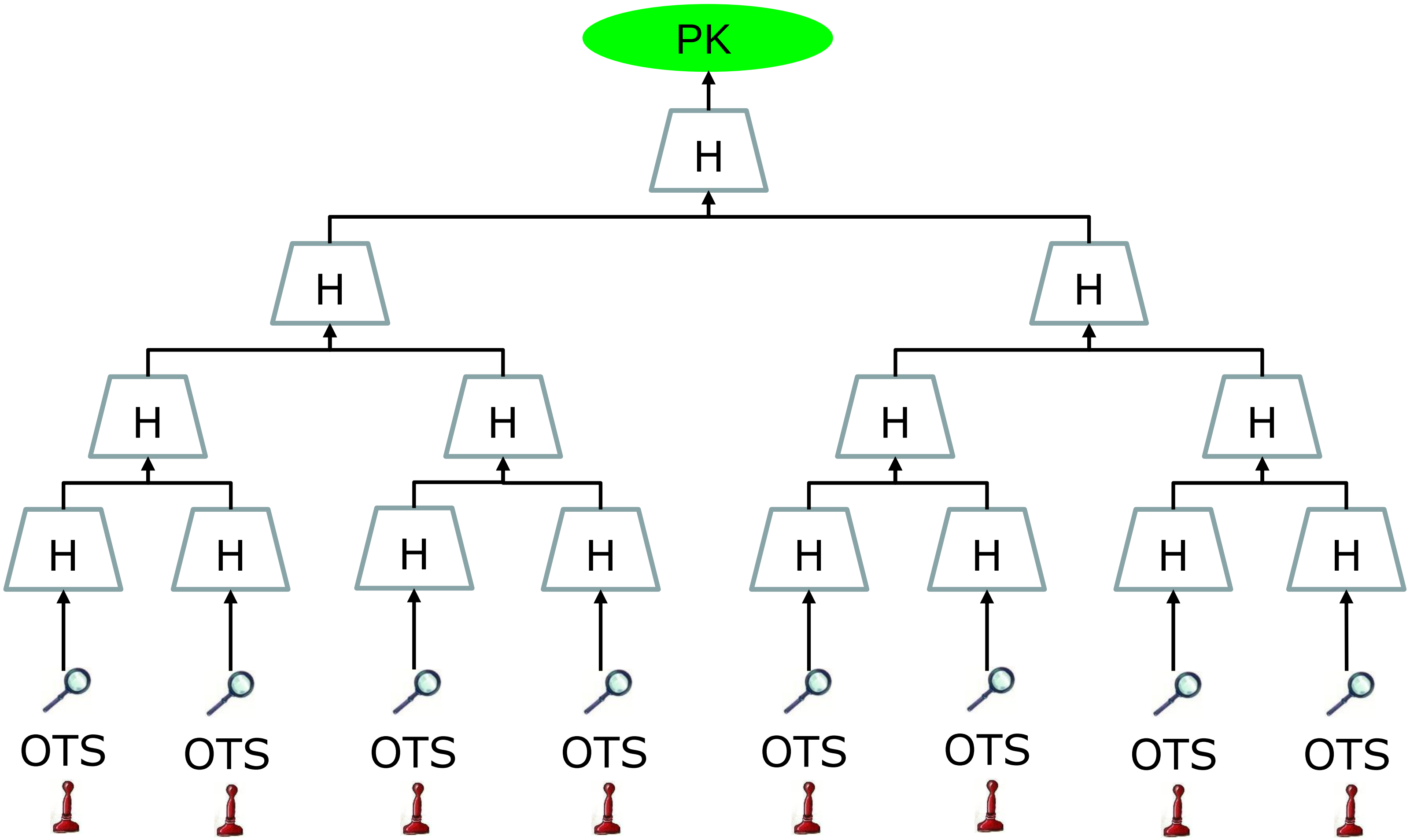
Merkle’s Hash-based Signatures



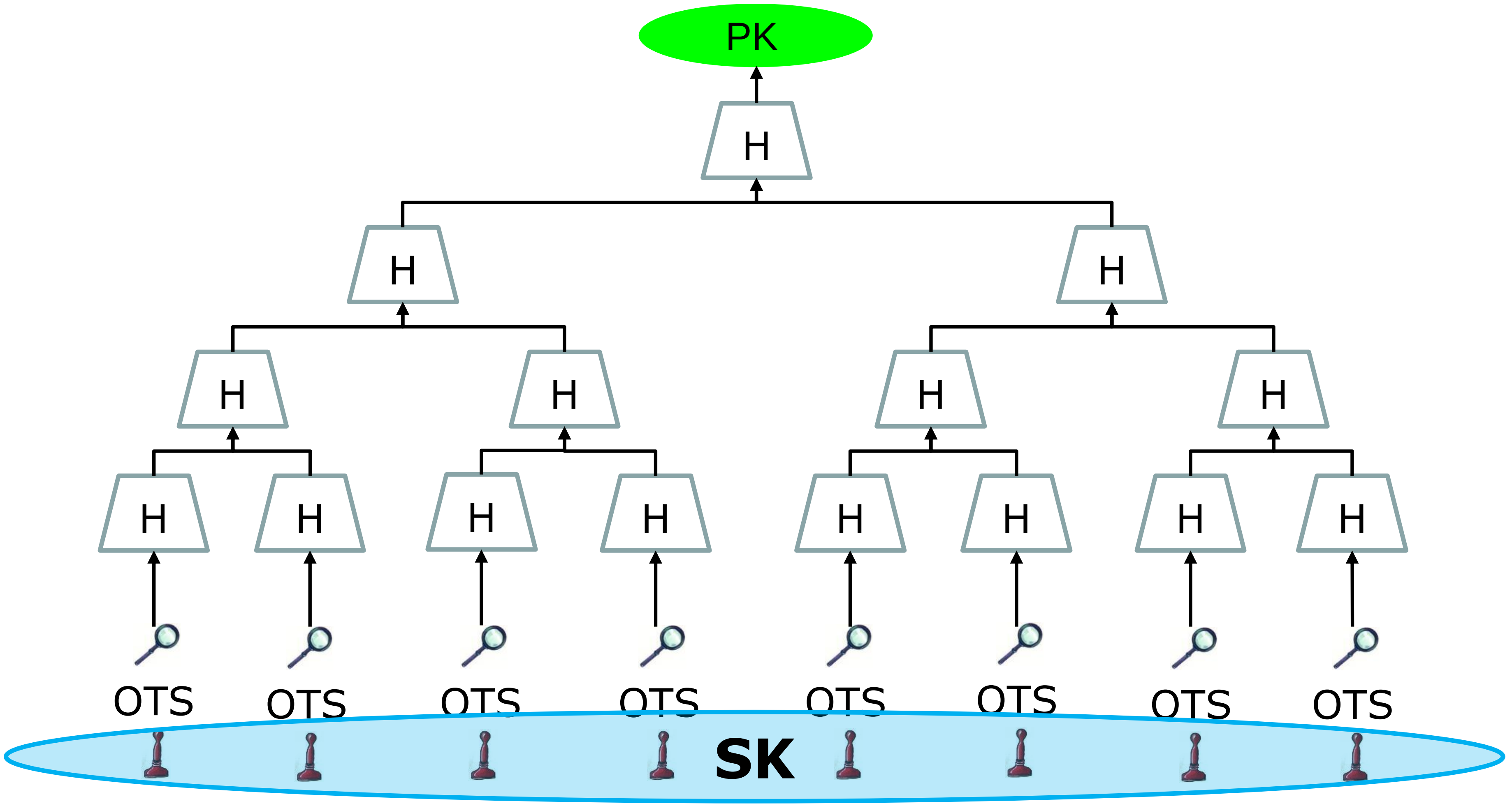
Merkle's Hash-based Signatures



Merkle's Hash-based Signatures

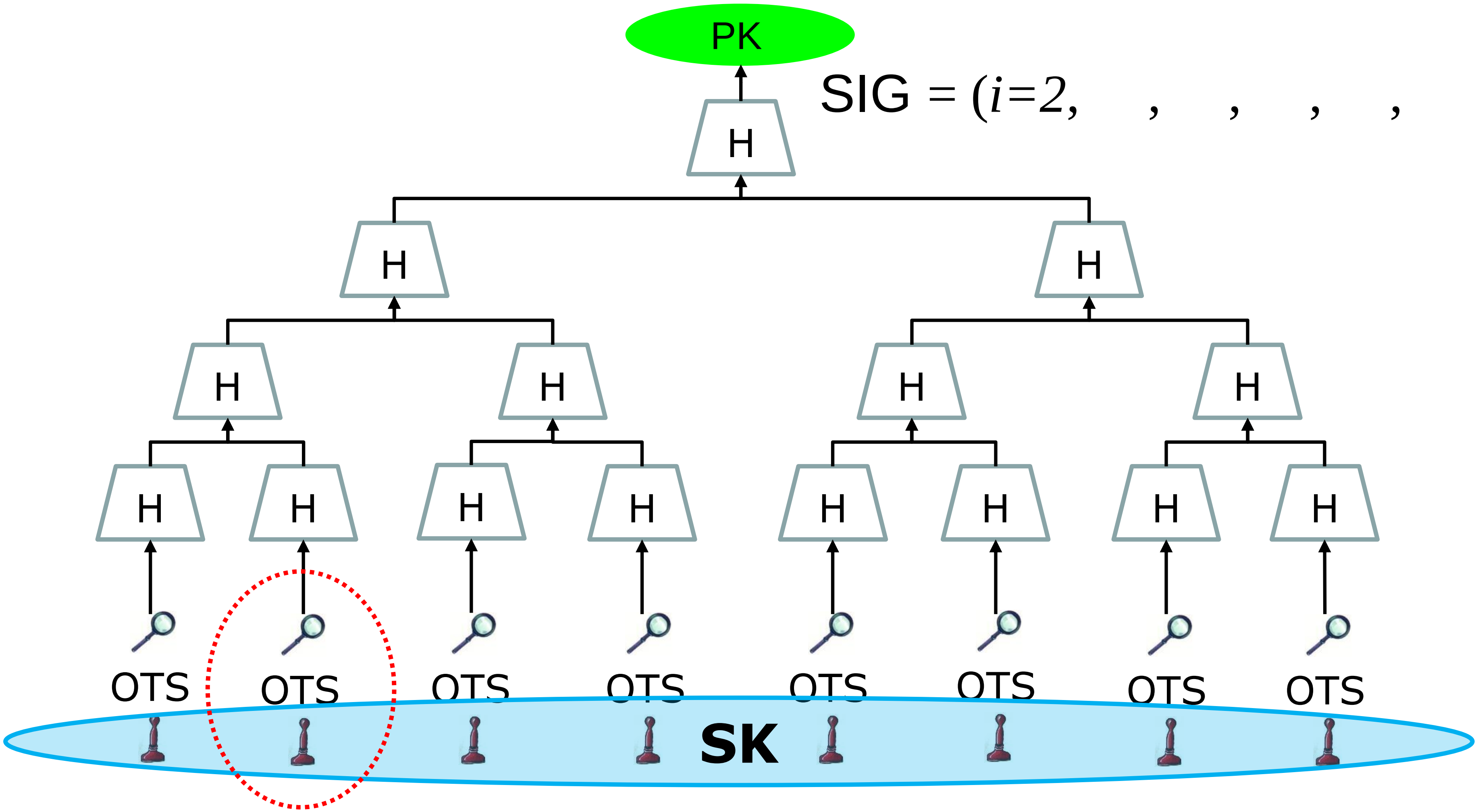


Merkle's Hash-based Signatures

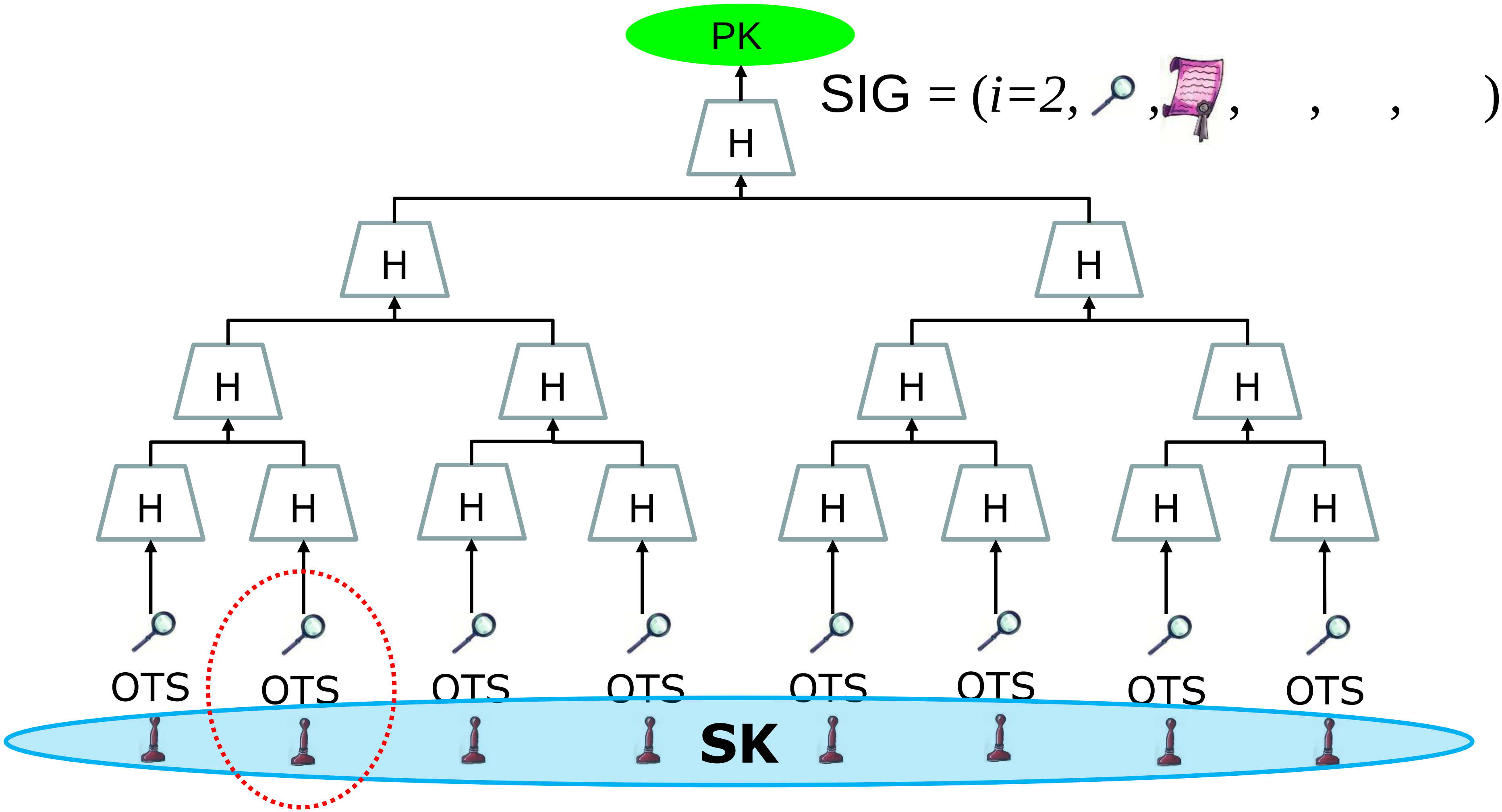


Merkle’s Hash-based Signatures

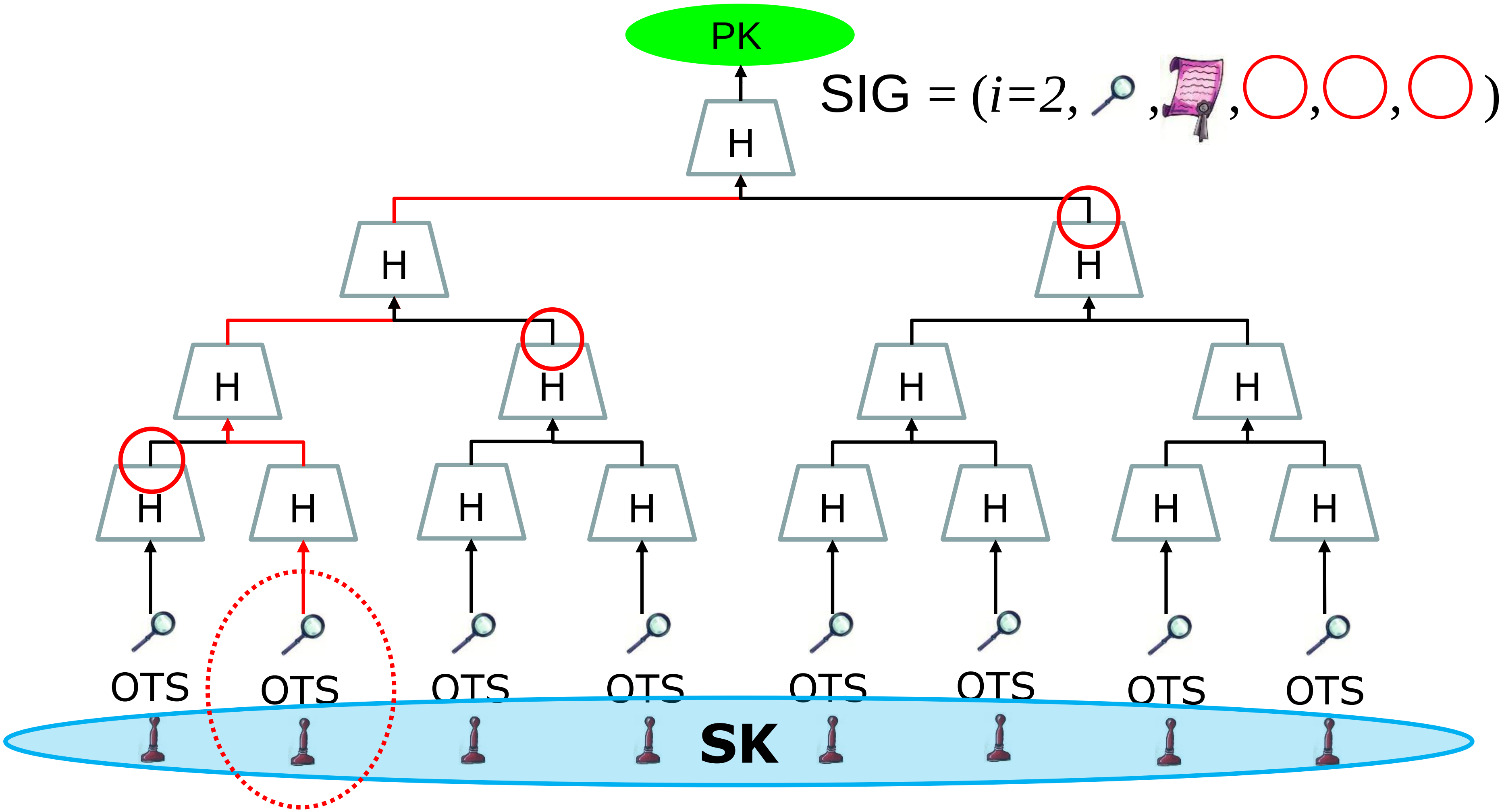
SIG = (i=2, , , , ,)



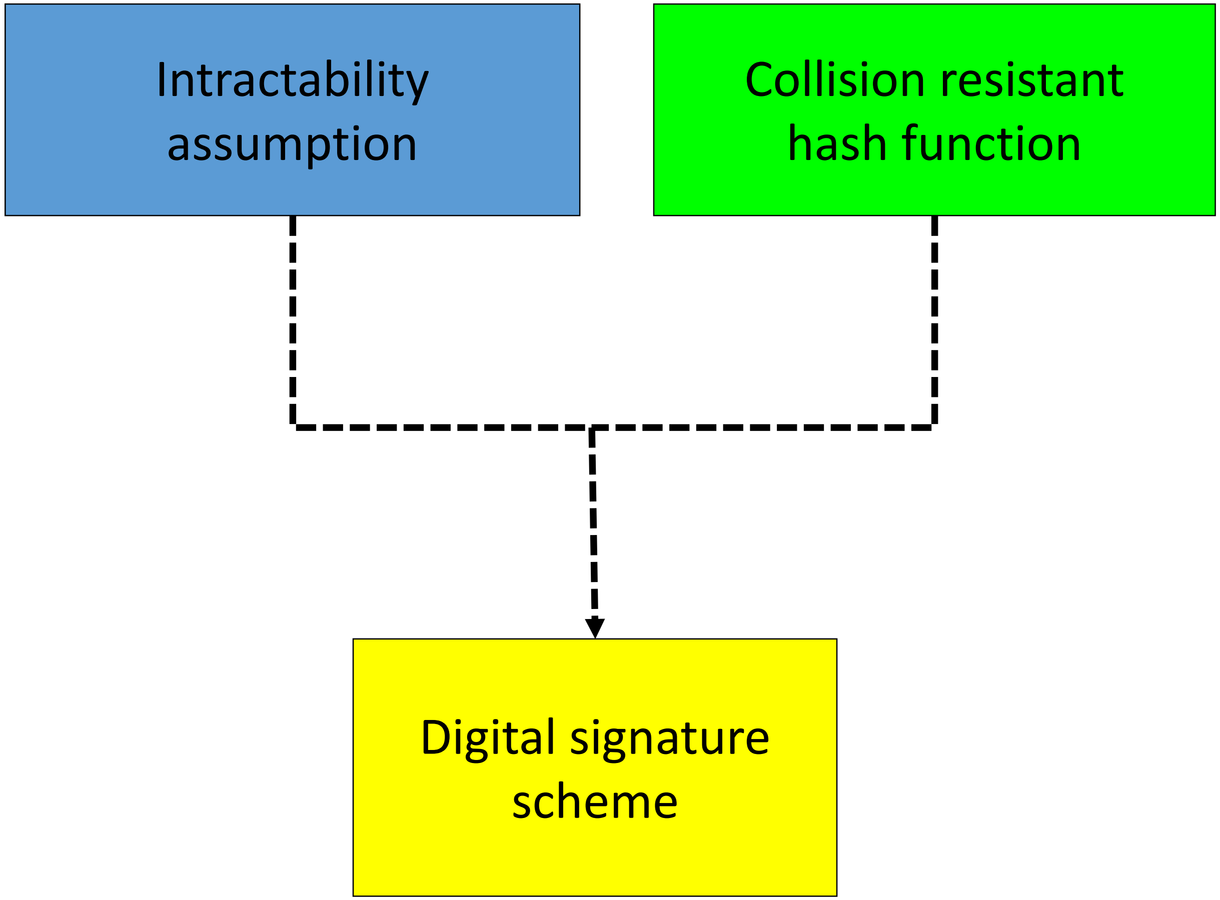
Merkle's Hash-based Signatures



Merkle's Hash-based Signatures



Security


Intractability assumption
Collision resistant hash function
Digital signature scheme

Post-Quantum Security

n-bit hash function

Grover'96:

Preimage finding $\boldsymbol{O(2^n) \rightarrow O(2^{\frac{n}{2}})}$

Brassard et al. 1998:

Collision finding $\boldsymbol{O(2^{\frac{n}{2}}) \rightarrow O(2^{\frac{n}{3}})}$

Aaronson & Shi'04:

Quantum collision finding $\boldsymbol{2^{\frac{n}{3}}}$ is lower bound

McGrew & Curcio‘2014

- Winternitz OTS (= LDWM-OTS)
- Merkle tree scheme (MTS)
- Parameter Sets = Cipher Suites
- Efficient sig / pk encoding
- Security <= collision resistance