Tweakable Hash Functions in Stateless Hash-Based Signature Schemes

Lena Heimberger

Graz, 16th January 2020

Mathematical Foundations of Cryptography

Motivation & Outline

hash functions are...

- well understood
- fast
- active research topic!
- e.g. signatures
- theoretical background
- proofs and notions

Contents

Hash-Based Signature Schemes

Hash-Based Signature Schemes[KF17]

- initially: Merkle Tree Signatures against Quantum Computers[Aug+17]
- minimal security assumptions
- One-Time Signatures
  - Lamport Signatures
  - WOTS/WOTS+
  - BiBa
  - ...
- Merkle Signatures
- XMSS and $XMSS^{MT}$
- stateful! keep track of all produced signatures

SPHINCS+[Aum+17]

- *Stateless Practical Hash-based Incredibly Nice Cryptographic Signatures*
- NIST Post-Quantum Project, Round 2
- small keys
- simple building blocks
- *stateless*: large structure (Goldreich)

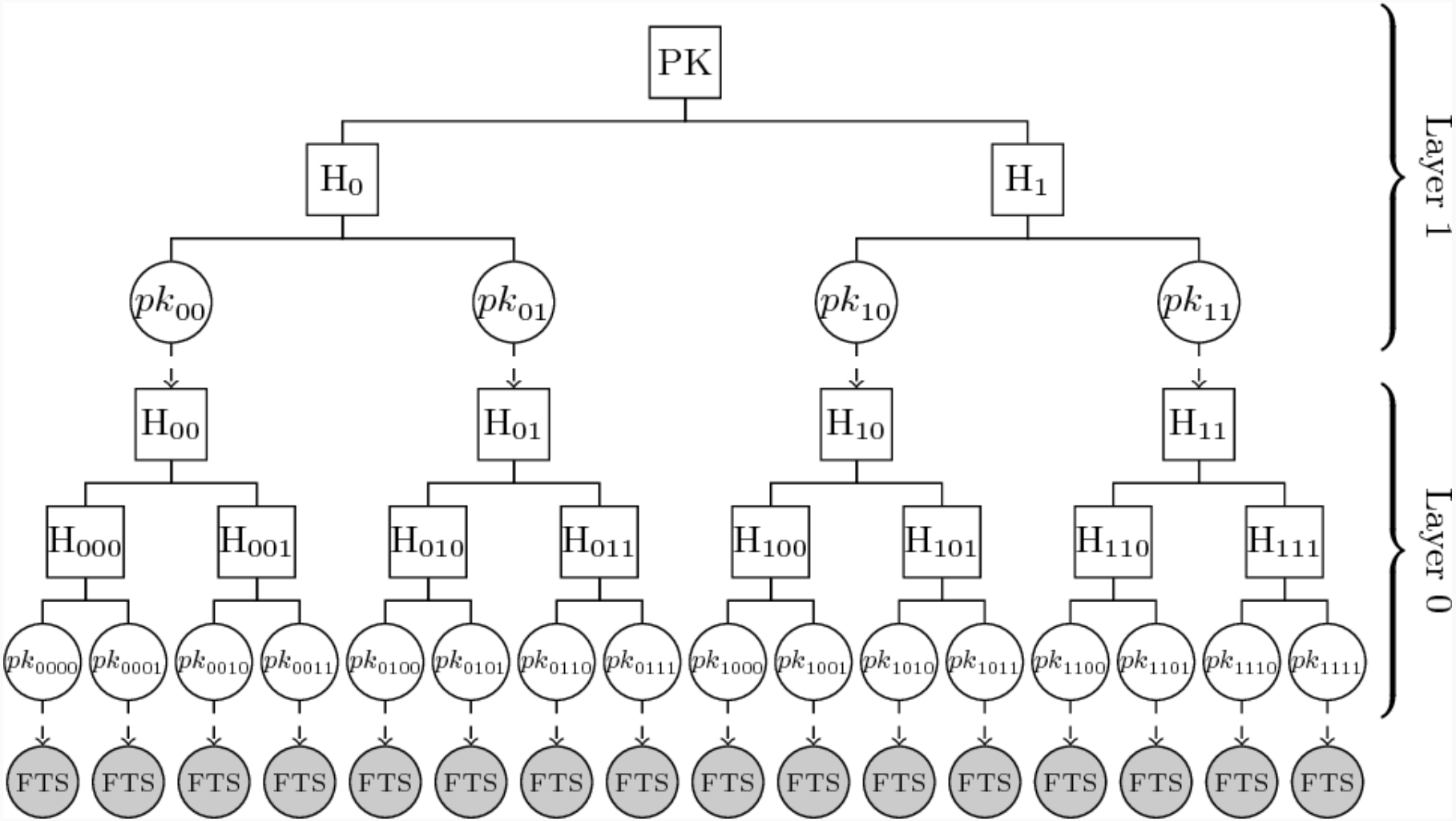


Figure 1: An overview of SPHINCS+, original picture from [CMP18]

Interlude: Proving Correctness [GHS15][Bin+19]

- **classical generic security**
  success probability against a random function family
- **quantum generic security**
  random function with superposition queries
- used primitive is still classical
- Random Oracle Model (ROM)
- Quantum Random Oracle Model (QROM)

Tweakable Hash Functions

Tweakable Hash Function[Ber+19]

- unify security analysis of hash-based signature schemes
- indifferent of used function
- independent calls
- three possible constructions

Tweakable Hash Function[Ber+19]

- *tweak* $T \in \mathcal{T}$
  context information, nonce
  `ADRS`
- *public parameters* $P \in \mathcal{P}$
  `PK.seed`
- $\mathbf{Th} : \mathcal{P} \times \mathcal{T} \times \{0,1\}^{\alpha} \to \{0,1\}^{n}$
  $MD \leftarrow \mathbf{Th}(P, T, M)$
- $\mathcal{P} = \{0,1\}^{n}$, $\mathcal{T} = \{0,1\}^{256}$
- Great, let's build one! (or three)

1^{st} approach to the construction of a Tweakable Hash Function

- standard model using a keyed hash function

$$H : \mathcal{K} \times \{0,1\}^{\alpha} \rightarrow \{0,1\}^{n}$$
$$\mathbf{Th}(P, T, M) = H(P_t, M^{\oplus})$$
$$P_t = P[(\alpha + n)T, n]$$
$$M^{\oplus} = M \oplus (P[(\alpha + n)T + n, \alpha])$$

- requires P to be linear to the size of the tweak space
- exponential-size public parameters: not suitable

2^{nd} approach to the construction of a Tweakable Hash Function[HRS16]

- two hash functions H_1 and H_2

$$H_1 : \{0,1\}^{2n} \times \{0,1\}^{\alpha} \rightarrow \{0,1\}^{n}$$
$$H_2 : \{0,1\}^{2n} \rightarrow \{0,1\}^{n}$$

2^{nd} approach to the construction of a Tweakable Hash Function[HRS16]

- two hash functions H_1 and H_2
- ~~public parameters~~ short public seed

$$H_1 : \{0,1\}^{2n} \times \{0,1\}^{\alpha} \rightarrow \{0,1\}^{n}$$
$$H_2 : \{0,1\}^{2n} \rightarrow \{0,1\}^{n}$$
$$\mathbf{Th}(P, T, M) = H_1(P||T, M^{\oplus})$$
$$M^{\oplus} = M \oplus H_2(P||T)$$

2^{nd} approach to the construction of a Tweakable Hash Function[HRS16]

- two hash functions H_1 and H_2
- ~~public parameters~~ short public seed

$$H_1 : \{0,1\}^{2n} \times \{0,1\}^{\alpha} \to \{0,1\}^{n}$$
$$H_2 : \{0,1\}^{2n} \to \{0,1\}^{n}$$
$$\mathbf{Th}(P, T, M) = H_1(P||T, M^{\oplus})$$
$$M^{\oplus} = M \oplus H_2(P||T)$$

- secure under the standard model
- require a QROM for public-parameter compression

3^{rd} approach to the construction of a Tweakable Hash Function

- assume all hash functions behave like QROs
- similar to LMS signatures-
  distinct prefix and suffix to avoid multi-collision attacks

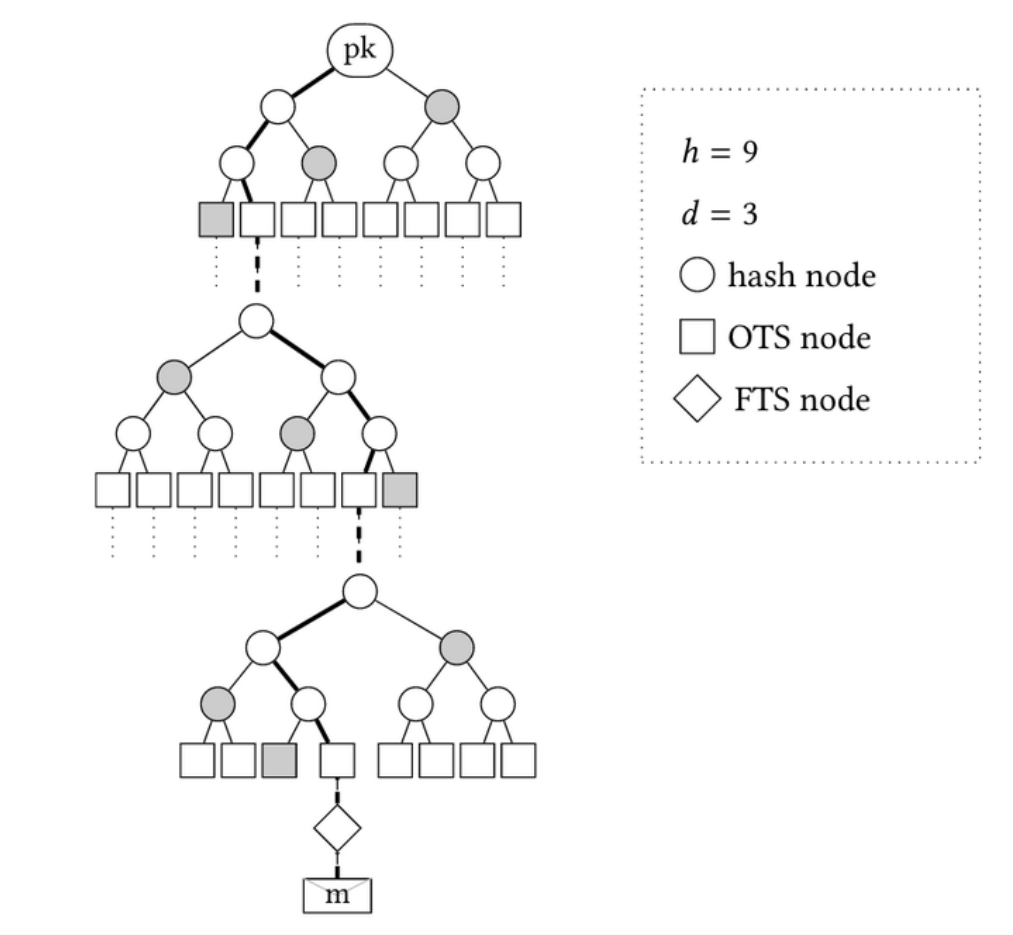
$$H : \{0,1\}^{2n+\alpha} \rightarrow \{0,1\}^n$$

$$\mathbf{Th}(P, T, M) = H(P||T||M)$$

Tweakable Hashes in SPHINCS+

Tweakable Hashes in SPHINCS+

- six different instantiantions
- using 2^{nd} or 3^{rd} approach
- SHA-256, SHAKE256 and Haraka
- *PQ-EU-CMA*



Cryptographic (Hash) Function Instantiation

- tweakable hash functions
  - limit message length to multiples of n
- pseudorandom functions
  - key and randomness generation
  - process messages of arbitrary length

Thank you for your attention!

If you have any questions, please do not hesitate to ask!

References

Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. "Semantic Security and Indistinguishability in the Quantum World". In: *IACR Cryptology ePrint Archive* 2015 (2015), p. 355. URL: https://eprint.iacr.org/2015/355.

Andreas Hülsing, Joost Rijneveld, and Fang Song. "Mitigating Multi-target Attacks in Hash-Based Signatures". In: *Public Key Cryptography (1)*. Springer, 2016, pp. 387–416. DOI: 10.1007/978-3-662-49384-7_15. URL: https://www.iacr.org/archive/pkc2016/96140179/96140179.pdf.

Daniel Augot, Lejla Batina, Daniel J. Bernstein, Joppe Bos, Johannes Buchmann, Wouter Castryck, Orr Dunkelmann, Tim Güneysu, Shay Gueron, Andreas Hülsing, Tanja Lange, Mohamed Saied Emam Mohamed, Christian Rechberger, Peter Schwabe, Nicolas Sendrier, Frederik Vercauteren, and Bo-Yin Yang. *LMS vs XMSS: Comparion of two Hash-Based Signature Standards*. Cryptology ePrint Archive, Report 2017/349. https://eprint.iacr.org/2017/349. 2017.

Jean Philippe Aumasson, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, and Peter Schwabe. "SPHINCS+". In: (Nov. 2017).

Panos Kampanakis and Scott Fluhrer. *LMS vs XMSS: Comparion of two Hash-Based Signature Standards*. Cryptology ePrint Archive, Report 2017/349. https://eprint.iacr.org/2017/349. 2017.

Laurent Castelnovi, Ange Martinelli, and Thomas Prest. *Grafting Trees: a Fault Attack against the SPHINCS framework*. Cryptology ePrint Archive, Report 2018/102. https://eprint.iacr.org/2018/102. 2018.

Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. *The SPHINCS+ Signature Framework*. Cryptology ePrint Archive, Report 2019/1086. https://eprint.iacr.org/2019/1086. 2019.

Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. "Tighter Proofs of CCA Security in the Quantum Random Oracle Model". In: Lecture Notes in Computer Science 11892 (2019), pp. 61–90. DOI: 10.1007/978-3-030-36033-7_3.