ARMed SPHINCS:

Computing a 41 KB signature in 16 KB of RAM

Andreas Hülsing[1], **Joost Rijneveld**[2], Peter Schwabe[2]

Technische Universiteit Eindhoven[1]
Radboud University, Nijmegen[2]
The Netherlands

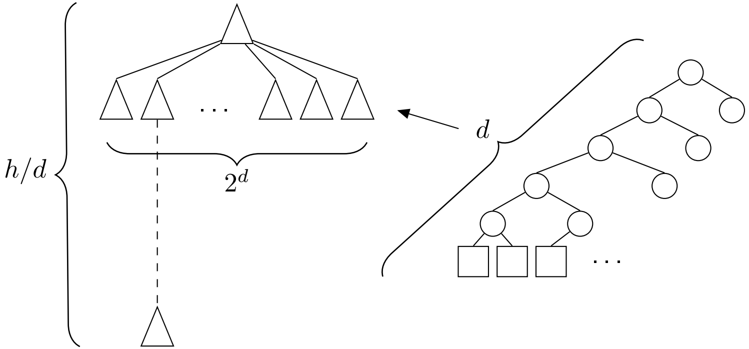
PQCRYPTO
ICT-645622

SPHINCS

- SPHINCS: Stateless, practical, hash-based, incredibly nice cryptographic signatures [BHHLNPSW15].
- Hash-based schemes: conservative choice
  - Hash functions do not fall to Shor (but halved by Grover)
  - One-way functions necessary for signatures [Rom90]
  - Tight security reductions
- SPHINCS-256 in PQCRYPTO's 'Initial recommendations'

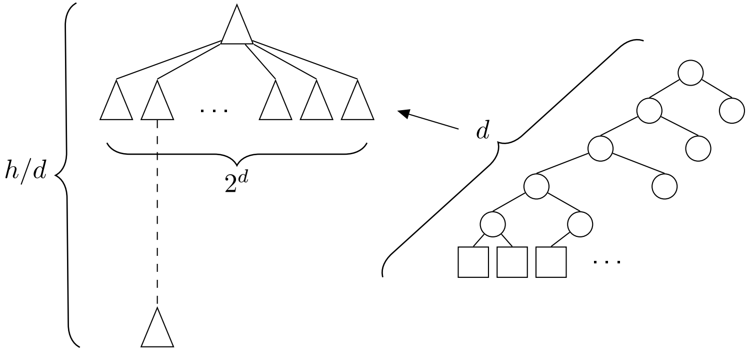
SPHINCS-256

- Large hash-tree, height $h = 60$
- Every $d = 12$-th layer: sign child node
  - Effectively a hypertree of $h/d = 5$ Merkle trees [Mer90]
  - Trade signature size for time
- Sign messages using 2^{60} leaf nodes



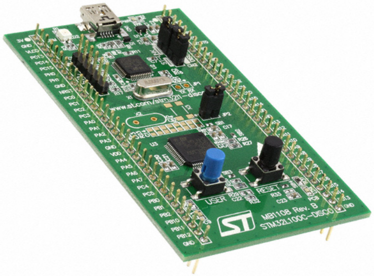
SPHINCS-256

- Large hash-tree, height $h = 60$
- Every $d = 12$-th layer: sign child node
  - Effectively a hypertree of $h/d = 5$ Merkle trees [Mer90]
  - Trade signature size for time
- Sign messages using 2^{60} leaf nodes
- No need to remember index: **stateless** [Gol87]



Platform

- STM32L100C development board
- Cortex M3, ARMv7-M
- libopencm3 firmware
- 32MHz, 32-bit architecture
- 16 registers
- 256KB Flash
- **16KB RAM**

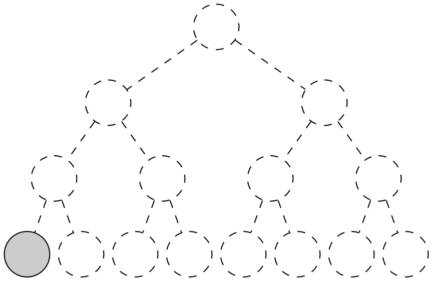


Treehash

- Some internal tree is too large: 2MB
  - 2^{16} leaf nodes

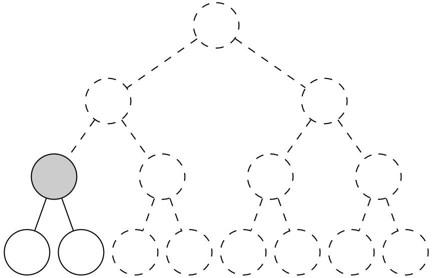
Treehash

- Some internal tree is too large: 2MB
  - 2^{16} leaf nodes
- Treehash [Mer90]: only store required nodes
  - Maintain a stack: at most $log(n) = 16$ nodes (or $log(8) = 3$, in the example below)



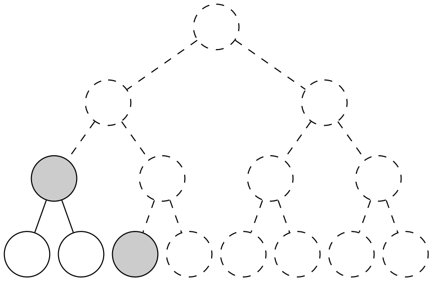
Treehash

- Some internal tree is too large: 2MB
  - 2^{16} leaf nodes
- Treehash [Mer90]: only store required nodes
  - Maintain a stack: at most $log(n) = 16$ nodes (or $log(8) = 3$, in the example below)



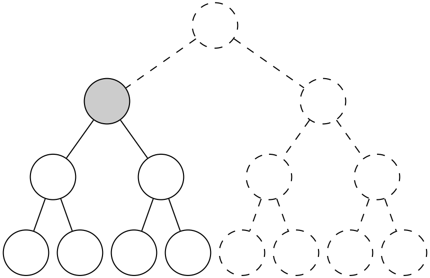
Treehash

- Some internal tree is too large: 2MB
  - 2^{16} leaf nodes
- Treehash [Mer90]: only store required nodes
  - Maintain a stack: at most $log(n) = 16$ nodes (or $log(8) = 3$, in the example below)



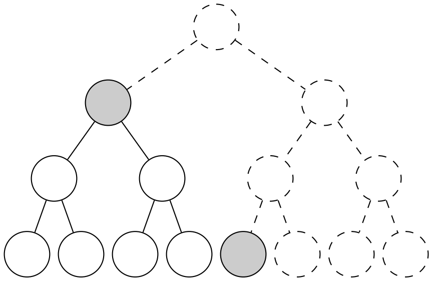
Treehash

- Some internal tree is too large: 2MB
  - 2^{16} leaf nodes
- Treehash [Mer90]: only store required nodes
  - Maintain a stack: at most $log(n) = 16$ nodes (or $log(8) = 3$, in the example below)



Treehash

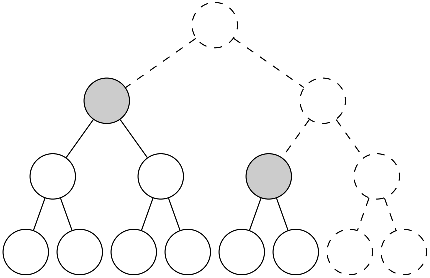
- Some internal tree is too large: 2MB
  - 2^{16} leaf nodes
- Treehash [Mer90]: only store required nodes
  - Maintain a stack: at most $log(n) = 16$ nodes (or $log(8) = 3$, in the example below)



PQCRYPTO
ICT-645622

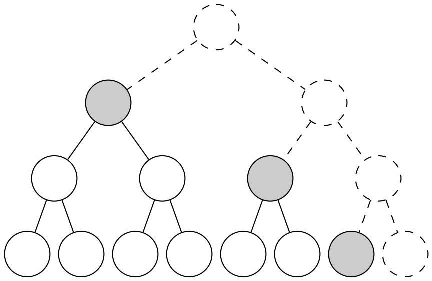
Treehash

- Some internal tree is too large: 2MB
  - 2^{16} leaf nodes
- Treehash [Mer90]: only store required nodes
  - Maintain a stack: at most $log(n) = 16$ nodes (or $log(8) = 3$, in the example below)



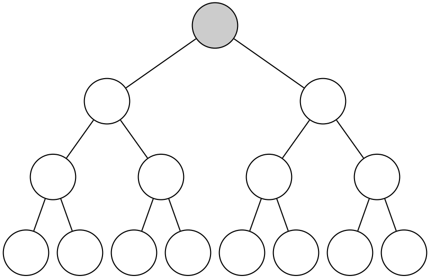
Treehash

- Some internal tree is too large: 2MB
  - 2^{16} leaf nodes
- Treehash [Mer90]: only store required nodes
  - Maintain a stack: at most $log(n) = 16$ nodes (or $log(8) = 3$, in the example below)



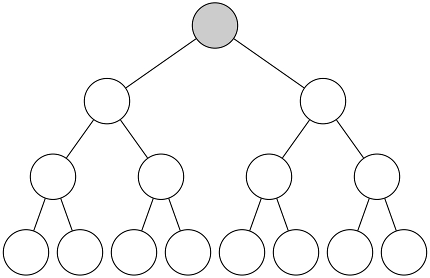
Treehash

- Some internal tree is too large: 2MB
  - 2^{16} leaf nodes
- Treehash [Mer90]: only store required nodes
  - Maintain a stack: at most $log(n) = 16$ nodes (or $log(8) = 3$, in the example below)



Treehash

- Some internal tree is too large: 2MB
    - 2^{16} leaf nodes
- Treehash [Mer90]: only store required nodes
    - Maintain a stack: at most $log(n) = 16$ nodes (or $log(8) = 3$, in the example below)
- Trace 32 ‘random’ paths through tree



Other aspects

- Streaming data
  - Signature does not fit in memory
    - ..and is generated out of order
  - Sizable messages do not fit in memory
  - Key material does not fit in memory
  - . . .
- Optimising ChaCha$_{12}$
  - 685818 calls per signature
  - 65% of total computation costs

Performance

- Works on 16KB RAM ✓
  - Uses less than 7KB

- Key generation: 0.88 seconds
- Signing: 18.41 seconds
- Verification: 0.51 seconds

Performance

- Works on 16KB RAM ✓
  - Uses less than 7KB

- Key generation: 0.88 seconds
- Signing: 18.41 seconds
- Verification: 0.51 seconds

- Implemented XMSS^{MT} [HRB13], configured similarly
  - Stateful: process leafs incrementally

Conclusions

- Stateless is expensive, but not prohibitively so
  - Signing 30x as expensive as XMSS^{MT}
  - Verification similar to XMSS^{MT}
  - (Key generation much cheaper)
- Feasible on limited platforms
  - ..although not very practical for interactive applications
- Code is available (public domain):
  https://joostrijneveld.nl/papers/armedsphincs/

Reference I

Daniel J. Bernstein, Diana Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Peter Schwabe and Zooko Wilcox O'Hearn.
SPHINCS: Stateless, practical, hash-based, incredibly nice cryptographic signatures.
In Marc Fischlin and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 368-397. Springer, 2015.

John Rompel.
One-way functions are necessary and sufficient for secure signatures.
In *Proceedings of the twenty-second annual ACM symposium on theory of computing*, pages 387–394. ACM, 1990.

Ralph Merkle.
A certified digital signature.
In Gilles Brassard, editor, *Advances in Cryptology – Crypto '89*, volume 435 of *LNCS*, pages 218-238. Springer-Verlag, 1990.

Reference II

Oded Goldreich.

Two remarks concerning the Goldwasser-Micali-Rivest signature scheme.

In Andrew M. Odlyzko, editor, *Advances in Cryptology – Crypto '86*, volume 263 of *LNCS*, pages 104-110. Springer-Verlag, 1987.

Andreas Hülsing, Lea Rausch and Johannes Buchmann.

Optimal Parameters for XMSS^{MT}.

In Alfredo Cuzzocrea, Christian Kittl, Dimitris E. Simos, Edgar Weippl and Lida Xu, editors, *Security Engineering and Intelligence Informatics*, volume 8128 of *LNCS*, pages 194-208. Springer, 2013.