XMSS: Extended Hash-Based Signatures
(draft-irtf-cfrg-xmss-hash-based-signatures-03)

A. Hülsing, D. Butin, S.-L. Gazdag, A. Mohaisen
Hash-based Signature Schemes

[Mer89]

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

FIG 1
An authentication tree with N = 8.
Merkle’s Hash-based Signatures

SIG = (i=2, H, OTS, SK, SK, SK, SK)
XMSS

Tree: Uses bitmasks

Leafs: Use binary tree with bitmasks

OTS: WOTS$^+$

Message digest: Randomized hashing

Collision-resilient

$\rightarrow$ 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
$\Theta(h/2) \rightarrow \Theta(h/2d)$
XMSS-T (Hülsing, Rijneveld, Song – PKC’16)

- draft-irtf-cfrg-xmss-hash-based-signatures actually implements XMSS-T not XMSS as published at PQCrypto’11
- Adds multi-target attack resistance
- Tight security reduction
  -> smaller sigs at same security

- Stateful, but building block for SPHINCS
Recent Changes
New Message Hash

Randomized hashing ($\text{dgst} = H(R_i, M_i)$) allows for Multi-Target-Attacks

- After $q$ signatures, find $(R, M)$ such that $H(R, M) = H(R_i, M_i)$ for $0 \leq i < q$
- Security level for $n$ bit hash function: $n - \log q$

Fix: Add index for domain separation

- 03 uses $\text{dgst} = H(R_i, i, M_i)$
- Prevents Multi-Target-Attacks in practice but no formal proof (but proof trivial in random oracle model).
Addressing Scheme

-02:
  • Fields were crossing byte and word boundaries
  • Annoying for implementers

-03:
  • Addresses redesigned to respect byte and word boundaries (where possible)
Upcoming changes

• Instantiation (used hash function)
• Addressing Scheme
• Generation of randomness for message hash
• Few more minor comments
Instantiation

• Currently:
  • SHA2-256 + ChaCha20 (mandatory)
  • SHA2-512 (mandatory)

• Discussion:
  • Adding SHA3 parameter sets? Optional or required?
  • Make SHA2-512 optional? (256 bit quantum security, 512 classical security)
  • Pure SHA2-256 as mandatory? (Code size / NIST support)
Instantiation

• Proposal:
  • SHA2-256 (mandatory)
    • Replace ChaCha20 by simplified HMAC construction (just prepend padded key, fine as dealing with fixed input size)
  • SHA2-512 (optional)
    • Same constructions as for SHA2-256
  • SHA3-256/512 (optional)
    • Proposal by van Assche / Daemon
    • Actually using SHAKE128 / SHAKE256
Addressing Scheme

• Introduces limits on parameter sets
• Critic: 40 bits for tree index not enough (indeed, not enough for SPHINCS)
• Address space currently exhausted
• Would need bigger addresses -> prevents use of ChaCha for key / bitmask generation -> speed penalty
Addressing Scheme

• Proposal:
  • Remove ChaCha20 instantiation
  • Increase address length to 32 bytes (currently 16 bytes)
  • Allows to assign sufficient space to all fields without crossing byte boundaries
Generation of R

- Currently „common approach“:
  \[ R = \text{PRF}(SK, M) \]

- As XMSS is stateful, we could do
  \[ R = \text{PRF}(SK, idx) \]
  + processing message just once
  - different from other schemes
Thank you!
Questions / Feedback ?