Hash-based Signatures: An outline for a new standard

A. Hülsing, D. Butin, S.-L. Gazdag
XMSS: Extended Hash-Based Signatures
(draft-huelsing-cfrg-hash-sig-xmss)
A. Hülsing, D. Butin, S.-L. Gazdag, A. Mohaisen
Hash-based Signature Schemes

Only secure hash function
Security well understood
Post quantum
Fast
Security

Intractability assumption

Collision resistant hash function

Digital signature scheme
Post-Quantum Security

n-bit hash function

Grover‘96:

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

Brassard et al. 1998:

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

Aaronson & Shi’04:

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

• Forward Secure Signatures
  • Security of old signatures after key compromise

• Delegatable / Proxy Signatures
  • Securely delegate signing rights

→ Require specific pseudorandom key gen
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

\[ \text{SIG} = (i=2, \ldots, \ldots, \ldots, \ldots, \ldots, ) \]
Merkle’s Hash-based Signatures

\[ \text{SIG} = (i=2, \text{OTS}, \text{OTS}, \ldots) \]
Merkle’s Hash-based Signatures

\[ \text{SIG} = (i=2, \mathcal{O}, \mathcal{O}, \mathcal{O}, \mathcal{O}) \]
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 (LDMW), 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\(^{\text{MT}}\))
General Design Choices

Define as mandatory:
• Public key and signature format & semantics
• Verification

Leave implementer freedom to choose trade-offs:
• 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\(^{\text{MT}}\) as subroutine

Efficient sig / pk encodings a la McGrew & Curcio
WOTS$^+$

Uses bitmasks
-> Collision-resilience
  -> signature size halved
-> Tighter security reduction

\[ H \quad \rightarrow \quad b_i \quad \rightarrow \quad H \]
XMSS

Tree: Uses bitmasks

Leafs: Use binary tree with bitmasks

OTS: WOTS

Message digest: Randomized hashing

-> Collision-resilience

-> Signature size halved
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

Same tree height and $w$ for all internal trees

$\rightarrow$ easier implementation
Design Choices: Parameters

Parameter sets for different settings

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

<table>
<thead>
<tr>
<th></th>
<th>$m = 256$, $n = 128$</th>
<th>$m = n = 256$</th>
<th>$m = n = 512$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical Security</td>
<td>128 bits</td>
<td>256 bits</td>
<td>512 bits</td>
</tr>
<tr>
<td>Post-Quantum Security</td>
<td>64 bits</td>
<td>128 bits</td>
<td>256 bits</td>
</tr>
<tr>
<td>Internal Hash</td>
<td>AES-128</td>
<td>SHA3-256</td>
<td>SHA3-512</td>
</tr>
<tr>
<td>Message Digest</td>
<td>SHA3-256</td>
<td>SHA3-256</td>
<td>SHA3-512</td>
</tr>
</tbody>
</table>
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?
McGrew & Curcio‘2014

• Winternitz OTS ( = LDWM-OTS)

• Merkle tree scheme (MTS)

• Parameter Sets = Cipher Suites

• Efficient sig / pk encoding

• Security <= collision resistance