Towards A Standard for Practical Hash-based Signatures

D. Butin, S.-L. Gazdag, A. Hülsing
Hash-based Signature Schemes [Mer89]

Post quantum

Only secure hash function

Security well understood

Fast
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
Merkle’s Hash-based Signatures

\[
\text{SIG} = (i=2, \text{OTS}, \text{OTS}, \text{OTS}, \text{OTS}, \text{OTS})
\]
Practical Challenge: Handle State

• Can be avoided in theory, paid with efficiency

• Different API
  • Handle Integration

• Prevent copies
  • No key back-up

• Multi-threading safety

• Industry input appreciated
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 (LDWLM), 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.
• Merkle Tree + Winternitz OTS

• Parameter Sets = Cipher Suites

• Security = collision resistance
XMSS

eXtended Merkle Signature Scheme
Reduced Security Requirements

- Change WOTS -> WOTS+
- Change Tree

Security from second-preimage resistance

„Collision-resilient“ scheme

No birthday-attacks
Size reduction

Hash function \( h: \{0,1\}^* \rightarrow \{0,1\}^m \)

Assume:
- only generic attacks,
- security level \( n \)

Collision resistance required:
\[ \rightarrow \text{generic attack = birthday attack} \quad \Rightarrow \quad m = 2n \]

Second-preimage resistance required:
\[ \rightarrow \text{generic attack = exhaustive search} \quad \Rightarrow \quad m = n \]
Early warning system

MD5 Collisions (theo.)

2004

SHA-1 Collisions (theo.)

2005

MD5 Collisions (practical!)

2008

MD5 & SHA-1
No (Second-) Preimage Attacks!

2014
Tree Chaining

Requires computation of $2 \times 2^{h/2}$ nodes in Merkle trees
Tree Chaining

• Can be extended to d layers

• Reduces signature and key generation time

• Necessary for smartcards & h >> 20
## Tree Chaining

<table>
<thead>
<tr>
<th></th>
<th>Sign (ms)</th>
<th>Verify (ms)</th>
<th>Keygen (ms)</th>
<th>Signature (byte)</th>
<th>Public Key (byte)</th>
<th>Secret Key (byte)</th>
<th>Bit Sec.</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMSS</td>
<td>134</td>
<td>23</td>
<td>925,400</td>
<td>2,388</td>
<td>800</td>
<td>2,448</td>
<td>92</td>
<td>$H = 16, w = 4$</td>
</tr>
<tr>
<td>XMSS+</td>
<td>106</td>
<td>25</td>
<td>5,600</td>
<td>3,476</td>
<td>544</td>
<td>3,760</td>
<td>94</td>
<td>$H = 16, w = 4$</td>
</tr>
<tr>
<td>RSA 2048</td>
<td>190</td>
<td>7</td>
<td>11,000</td>
<td>$\leq 256$</td>
<td>$\leq 512$</td>
<td>$\leq 512$</td>
<td>87</td>
<td></td>
</tr>
</tbody>
</table>

Infineon SLE78 16Bit-CPU@33MHz, 8KB RAM, TRNG, sym. & asym. co-processor

NVM: Card 16.5 million write cycles/ sector,
XMSS+ < 5 million write cycles (h=20)

[HBB12]
Forward Security

**Goal:** \( \text{Sig} = (\sigma, j), j < i \)
Requires special KeyGen
C Implementation, using OpenSSL [BDH2011]

<table>
<thead>
<tr>
<th></th>
<th>Sign (ms)</th>
<th>Verify (ms)</th>
<th>Signature (bit)</th>
<th>Public Key (bit)</th>
<th>Secret Key (byte)</th>
<th>Bit Security</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMSS-SHA-2</td>
<td>35.60</td>
<td>1.98</td>
<td>16,672</td>
<td>13,600</td>
<td>3,364</td>
<td>157</td>
<td>h = 20, w = 64,</td>
</tr>
<tr>
<td>XMSS-AES-NI</td>
<td>0.52</td>
<td>0.07</td>
<td>19,616</td>
<td>7,328</td>
<td>1,684</td>
<td>84</td>
<td>h = 20, w = 4</td>
</tr>
<tr>
<td>XMSS-AES</td>
<td>1.06</td>
<td>0.11</td>
<td>19,616</td>
<td>7,328</td>
<td>1,684</td>
<td>84</td>
<td>h = 20, w = 4</td>
</tr>
<tr>
<td>RSA 2048</td>
<td>3.08</td>
<td>0.09</td>
<td>≤ 2,048</td>
<td>≤ 4,096</td>
<td>≤ 512</td>
<td>87</td>
<td></td>
</tr>
</tbody>
</table>

**Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz with Intel AES-NI**
Conclusion

• Current draft: Great first step

... BUT ...

• XMSS: Additional important features
  • More efficient
  • Stronger Security Guarantees
  • Forward-security

Add-on to draft required.
Thank you!
Questions?