Hash-based Signature, the Round-3 Candidate: SPHINCS+

Tanja Lange

Academia Sinica

Eindhoven University of Technology

Post-quantum cryptography forum
14 January 2022
Public-key signatures

Prerequisite: Alice has a private key 🔑 and public key 🚀.
Prerequisite: Everyone knows 🚀 as belonging to Alice.
Alice signs messages using 🔑. Other people verify using 🚀.

Security goals: Integrity and authenticity. Nobody can produce signatures valid under 🚀 without 🚀. Modifications to signed message get caught.
Prerequisite: Alice has a private key and public key.
Prerequisite: Everyone knows as belonging to Alice.
Alice signs messages using . Other people verify using .
Security goals: Integrity and authenticity.
Nobody can produce signatures valid under without .
Modifications to signed message get caught.
第一屆後量子密碼論壇

Post-quantum Cryptography Forum
Connection Security for pqc.ithome.com.tw

You are securely connected to this site.

Verified by: TAIWAN-CA
Website Identity
Website: pqc.ithome.com.tw
Owner: This website does not supply ownership information.
Verified by: TAIWAN-CA
Expires on: January 3, 2023

Privacy & History
Have I visited this website prior to today? Yes, once
Is this website storing information on my computer? Yes, cookies
Have I saved any passwords for this website? No

Technical Details
Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, 128 bit keys, TLS 1.2)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.
<table>
<thead>
<tr>
<th>Certificate Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subject Name</strong></td>
</tr>
<tr>
<td>Country</td>
</tr>
<tr>
<td>State/Province</td>
</tr>
<tr>
<td>Locality</td>
</tr>
<tr>
<td>Organization</td>
</tr>
<tr>
<td>Organizational Unit</td>
</tr>
<tr>
<td>Common Name</td>
</tr>
</tbody>
</table>

| **Issuer Name** |
| Country | TW |
| Organization | TAIWAN-CA |
| Organizational Unit | Secure SSL Sub-CA |
| Common Name | TWCA Secure SSL Certification Authority |

| **Validity** |
| Not Before | Thu, 16 Dec 2021 08:43:55 GMT |
| Not After | Tue, 03 Jan 2023 15:59:59 GMT |

| **Subject Alt Names** |
| DNS Name | *.ithome.com.tw |
| DNS Name | ithome.com.tw |

| **Public Key Info** |
| Algorithm | RSA |
| Key Size | 2048 |
| Exponent | 65537 |
| Modulus | 0x51:00:00:00:00:04:EA:45:4A:58:85:3B:12:01:31 |

| **Miscellaneous** |
| Serial Number | 4F:35:00:00:00:04:EA:15:4A:58:85:3B:12:01:31 |
| Signature Algorithm | SHA-256 with RSA Encryption |
## Public Key Info

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Size</td>
<td>2048</td>
</tr>
<tr>
<td>Exponent</td>
<td>65537</td>
</tr>
</tbody>
</table>

## Miscellaneous

<table>
<thead>
<tr>
<th>Serial Number</th>
<th>47:E5:00:00:00:04:EA:15:4A:58:85:C3:B1:2D:1A:51</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature Algorithm</td>
<td>SHA-256 with RSA Encryption</td>
</tr>
<tr>
<td>Version</td>
<td>3</td>
</tr>
<tr>
<td>Download</td>
<td>PEM (cert) PEM (chain)</td>
</tr>
</tbody>
</table>

## Fingerprints

|------------|---------------------------------------------------------------------|
Post-quantum public-key signatures: hash-based

- Only one prerequisite: a good hash function, e.g. SHA3-512, …
  Hash functions map long strings to fixed-length strings.
  \( H : \{0, 1\}^* \to \{0, 1\}^n. \)

  Signature schemes use hash functions in handling.

- Old idea: 1979 Lamport one-time signatures;
  1979 Merkle extends to more signatures.
Post-quantum public-key signatures: hash-based

Only one prerequisite: a good hash function, e.g. SHA3-512, ... Hash functions map long strings to fixed-length strings.

$H : \{0, 1\}^* \rightarrow \{0, 1\}^n$.

Signature schemes use hash functions in handling...

- Quantum computers affect the hardness only marginally (Grover, not Shor).
- Old idea: 1979 Lamport one-time signatures; 1979 Merkle extends to more signatures.
One-time signatures (Lamport and Winternitz)

Idea: Use one-wayness of cryptographic hash function to authenticate.
Private key: bit string $s$, public key: $H(s)$.
One-time signatures (Lamport and Winternitz)

Idea: Use one-wayness of cryptographic hash function to authenticate.
Private key: bit string $s$, public key: $H(s)$.
Can only use once as $s$ known after first use.
One-time signatures (Lamport and Winternitz)

Idea: Use one-wayness of cryptographic hash function to authenticate.
Private key: bit string $s$, public key: $H(s)$.
Can only use once as $s$ known after first use.

Extend to signing bit by having two values:
Private key: 2 bit strings $(s_0, s_1)$, public key: $(H(s_0), H(s_1))$. 
One-time signatures (Lamport and Winternitz)

Idea: Use one-wayness of cryptographic hash function to authenticate.
Private key: bit string $s$, public key: $H(s)$.
Can only use once as $s$ known after first use.

Extend to signing bit by having two values:
Private key: 2 bit strings $(s_0, s_1)$, public key: $(H(s_0), H(s_1))$.
To sign 0 reveal $s_0$, to sign 1 reveal $s_1$. 
One-time signatures (Lamport and Winternitz)

Idea: Use one-wayness of cryptographic hash function to authenticate.
Private key: bit string $s$, public key: $H(s)$.
Can only use once as $s$ known after first use.

Extend to signing bit by having two values:
Private key: 2 bit strings ($s_0$, $s_1$), public key: ($H(s_0)$, $H(s_1)$).
To sign 0 reveal $s_0$, to sign 1 reveal $s_1$.

Lamport signs $m$ via $H(m) = (h_0, h_1, \ldots, h_{255})$.
Private key: $256 \times 2$ bit strings $s = (s_{0,0}, s_{0,1}, s_{1,0}, s_{1,1}, \ldots, s_{255,0}, s_{255,1})$,
public key: $p = (H(s_{0,0}), H(s_{0,1}), H(s_{1,0}), H(s_{1,1}), \ldots, H(s_{255,0}), H(s_{255,1}))$.
To sign $m$ reveal $s_{0,h_0}, s_{1,h_1}, \ldots, s_{255,h_{255}}$.
One-time signatures (Lamport and Winternitz)

Idea: Use one-wayness of cryptographic hash function to authenticate.
Private key: bit string \( s \), public key: \( H(s) \).
Can only use once as \( s \) known after first use.

Extend to signing bit by having two values:
Private key: 2 bit strings \((s_0, s_1)\), public key: \((H(s_0), H(s_1))\).
To sign 0 reveal \( s_0 \), to sign 1 reveal \( s_1 \).

Lamport signs \( m \) via \( H(m) = (h_0, h_1, \ldots, h_{255}) \).
Private key: \( 256 \times 2 \) bit strings \( s = (s_{0,0}, s_{0,1}, s_{1,0}, s_{1,1}, \ldots, s_{255,0}, s_{255,1}) \),
public key: \( p = (H(s_{0,0}), H(s_{0,1}), H(s_{1,0}), H(s_{1,1}), \ldots, H(s_{255,0}), H(s_{255,1})) \).
To sign \( m \) reveal \( s_{0,h_0}, s_{1,h_1}, \ldots, s_{255,h_{255}} \).
Tradeoff: define public key as \( H(p) \), also reveal rest of \( p \) to sign,
for short public key at expense of longer signature.
One-time signatures (Lamport and Winternitz)

Idea: Use one-wayness of cryptographic hash function to authenticate.
Private key: bit string $s$, public key: $H(s)$.
Can only use once as $s$ known after first use.

Extend to signing bit by having two values:
Private key: 2 bit strings $(s_0, s_1)$, public key: $(H(s_0), H(s_1))$.
To sign 0 reveal $s_0$, to sign 1 reveal $s_1$.

Lamport signs $m$ via $H(m) = (h_0, h_1, \ldots, h_{255})$.
Private key: $256 \times 2$ bit strings $s = (s_{0,0}, s_{0,1}, s_{1,0}, s_{1,1}, \ldots, s_{255,0}, s_{255,1})$,
public key: $p = (H(s_{0,0}), H(s_{0,1}), H(s_{1,0}), H(s_{1,1}), \ldots, H(s_{255,0}), H(s_{255,1}))$.
To sign $m$ reveal $s_{0,h_0}, s_{1,h_1}, \ldots, s_{255,h_{255}}$.
Tradeoff: define public key as $H(p)$, also reveal rest of $p$ to sign,
for short public key at expense of longer signature.

Winternitz achieves short public keys and signatures costing more calls to $H$. 

Tanja Lange
Hash-based Signature, the Round-3 Candidate: SPHINCS+
On the fast track: stateful hash-based signatures

- CFRG has published 2 RFCs: RFC 8391 and RFC 8554

---

**RFC 8391**

- **Request for Comments**: 8391
- **Category**: Informational
- **ISSN**: 2070-1721

*Internet Research Task Force (IRTF)*

XMSS: eXtended Merkle Signature Scheme

- **A. Huelsing**
- **TU Eindhoven**
- **D. Butin**
- **TU Darmstadt**
- **S. Gazdag**
- **genua GmbH**
- **J. Rijneveld**
- **Radboud University**
- **A. Mohaisen**
- University of Central Florida
- **May 2018**

---

**RFC 8554**

- **Request for Comments**: 8554
- **Category**: Informational
- **ISSN**: 2070-1721

*Internet Research Task Force (IRTF)*

Leighton-Micali Hash-Based Signatures

- **D. McGrew**
- **M. Curcio**
- **S. Fluhrer**
- **Cisco Systems**
- **April 2019**
On the fast track: stateful hash-based signatures

- CFRG has published 2 RFCs: RFC 8391 and RFC 8554
- NIST has standardized the same two schemes.
On the fast track: stateful hash-based signatures

- CFRG has published 2 RFCs: RFC 8391 and RFC 8554
- NIST has standardized the same two schemes.

- ISO SC27 JTC1 WG2 is working on standard for stateful hash-based signatures.
Merkle’s (e.g.) 8-time signature system

Hash 8 one-time public keys into a single Merkle public key \( P_{15} \).

\[
P_{15} = H(P_{13}, P_{14})
\]

\[
P_{13} = H(P_9, P_{10})
\]

\[
P_9 = H(P_1, P_2)
\]

\[
P_{14} = H(P_{11}, P_{12})
\]

\[
P_{10} = H(P_3, P_4)
\]

\[
P_{11} = H(P_5, P_6)
\]

\[
P_{12} = H(P_7, P_8)
\]

\[
P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5 \quad P_6 \quad P_7 \quad P_8
\]

\[
S_1 \quad S_2 \quad S_3 \quad S_4 \quad S_5 \quad S_6 \quad S_7 \quad S_8
\]

\( S_i \rightarrow P_i \) can be Lamport or Winternitz one-time signature system.

Each such pair \((S_i, P_i)\) may be used only once.
Signature in 8-time Merkle hash tree

Signature of first message: \((\text{sign}(m, S_1), P_1, P_2, P_{10}, P_{14})\).

\[ P_{15} = H(P_{13}, P_{14}) \]

\[ P_{13} = H(P_9, P_{10}) \]

\[ P_9 = H(P_1, P_2) \]

\[ P_{10} = H(P_3, P_4) \]

\[ P_{11} = H(P_5, P_6) \]

\[ P_{12} = H(P_7, P_8) \]

Verify signature \(\text{sign}(m, S_1)\) with public key \(P_1\) (provided in signature).

Link \(P_1\) against public key \(P_{15}\) by computing \(P'_9 = H(P_1, P_2)\), \(P'_{13} = H(P'_9, P_{10})\), and comparing \(H(P'_{13}, P_{14})\) with \(P_{15}\). Reject if \(H(P'_{13}, P_{14}) \neq P_{15}\).
Signature in 8-time Merkle hash tree

Signature of first message: \((\text{sign}(m, S_1), P_1, P_2, P_{10}, P_{14})\).

\[ P_{15} = H(P_{13}, P_{14}) \]

\[ P_{13} = H(P_9, P_{10}) \]

\[ P_9 = H(P_1, P_2) \]

\[ P_3 \]

\[ P_2 \]

\[ P_1 \]

\[ S_1 \]

\[ P_4 \]

\[ P_{10} = H(P_3, P_4) \]

\[ P_3 \]

\[ P_7 \]

\[ P_8 \]

\[ P_10 \]

\[ P_9 \]

\[ P_5 \]

\[ P_6 \]

\[ S_3 \]

\[ S_4 \]

\[ S_7 \]

\[ S_8 \]

\[ P_5 \]

\[ P_6 \]

\[ P_7 \]

\[ P_8 \]

\[ S_5 \]

\[ S_6 \]

\[ S_7 \]

\[ S_8 \]

Verify signature \(\text{sign}(m, S_1)\) with public key \(P_1\) (provided in signature).

Link \(P_1\) against public key \(P_{15}\) by computing \(P'_9 = H(P_1, P_2)\), \(P'_{13} = H(P'_9, P_{10})\), and comparing \(H(P'_{13}, P_{14})\) with \(P_{15}\). Reject if \(H(P'_{13}, P_{14}) \neq P_{15}\).
Huge trees (1987 Goldreich), keys on demand (Levin)

Signer chooses random \( r \in \{2^{255}, 2^{255} + 1, \ldots, 2^{256} - 1\} \), uses one-time public key \( T_r \) to sign \( m \); uses one-time public key \( T_i \) to sign \((T_{2i}, T_{2i+1})\) on path to \( T_1 \).
Generates \( i \)th secret key deterministically as \( H_k(i) \) where \( k \) is master secret.

Important for efficiency
Huge trees (1987 Goldreich), keys on demand (Levin)

Signer chooses random \( r \in \{2^{255}, 2^{255} + 1, \ldots, 2^{256} - 1\} \), uses one-time public key \( T_r \) to sign \( m \); uses one-time public key \( T_i \) to sign \((T_{2i}, T_{2i+1})\) on path to \( T_1 \).

Generates \( i \)th secret key deterministically as \( H_k(i) \) where \( k \) is master secret.

Important for efficiency

\[
\begin{array}{c}
T_1 \\
\Downarrow \\
T_2 \\
\Downarrow \\
T_3 \\
\Downarrow \\
\vdots \\
\Downarrow \\
T_r \\
\Downarrow \\
m \\
\end{array}
\]

\( T_i \) for small \( i \) gets used repeatedly (each time an \( m \) falls in that sub-tree)
Huge trees (1987 Goldreich), keys on demand (Levin)

Signer chooses random $r \in \{2^{255}, 2^{255} + 1, \ldots, 2^{256} - 1\}$, uses one-time public key $T_r$ to sign $m$; uses one-time public key $T_i$ to sign $(T_{2i}, T_{2i+1})$ on path to $T_1$. Generates $i$th secret key deterministically as $H_k(i)$ where $k$ is master secret.

Important for efficiency

$$\begin{array}{c}
T_1 \\
T_2 \\
T_3 \\
\vdots \\
T_{2^{256}} \\
\vdots \\
T_{2^{255} - 1} \\
m
\end{array}$$

$t_i$ for small $i$ gets used repeatedly (each time an $m$ falls in that sub-tree) but $H_k(i)$ being deterministic means it signs the same value, so no break.
Huge trees (1987 Goldreich), keys on demand (Levin)

Signer chooses random \( r \in \{2^{255}, 2^{255}+1, \ldots, 2^{256} - 1\} \), uses one-time public key \( T_r \) to sign \( m \); uses one-time public key \( T_i \) to sign \((T_{2i}, T_{2i+1})\) on path to \( T_1 \).
Generates \( i \)th secret key deterministically as \( H_k(i) \) where \( k \) is master secret.
Important for efficiency and security.

\[
\begin{align*}
    T_1 & \quad \rightarrow \\
    T_2 & \quad \rightarrow \\
    T_3 & \\
    \downarrow & \\
    \vdots & \\
    T_{2^{254}} & \quad \rightarrow \\
    T_{2^{255}} & \quad \rightarrow \\
    T_{2^{256}-2} & \quad \rightarrow \\
    T_{2^{256}-1} & \\
    m & \quad \downarrow \\
\end{align*}
\]

\( T_i \) for small \( i \) gets used repeatedly (each time an \( m \) falls in that sub-tree) but \( H_k(i) \) being deterministic means it signs the same value, so no break.
NIST submission SPHINCS+

- Post-quantum signature based on hash functions.
- Requires only a secure hash function, no further assumptions.
- Based on ideas of Lamport (1979) and Merkle (1979).
- Developed starting from SPHINCS with
  - improve multi-signature,
  - smaller keys,
  - Option for shorter signatures (30kB instead of 41kB) if “only” $2^{50}$ messages signed.
- Three versions (using different hash functions)
  - SPHINCS+-SHA3 (with SHAKE256),
  - SPHINCS+-SHA2 (with SHA-256),
  - SPHINCS+-Haraka (with Haraka, a hash function for short inputs).

More info at https://sphincs.org/.

See also my course page for more detailed videos and slides.