SPHINCS$^+$

Jean-Philippe Aumasson, Daniel J. Bernstein, Ward Beullens, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Bas Westerbaan
Hash-based signatures
(Merkle ‘89)

Boring crypto:

• Dates back to beginning of public key cryptography
• No fancy new mathematical assumption: Only requires a secure hash function („minimal security assumptions“)
• Stateful schemes already in standardization
Hash-based signatures
(Merkle ‘89)

Boring crypto:

• Dates back to beginning of public key cryptography
• No fancy new mathematical assumption: Only requires a secure hash function („minimal security assumptions“)
• Stateful schemes already in standardization
SPHINCS (Eurocrypt 2015)

Joint work with Daniel J. Bernstein, Daira Hopwood, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O’Hearn

https://sphincs.org/
Stateless hash-based signatures
[NY89,Gol87,Gol04]

Goldreich’s approach [Gol04]:
Security parameter $\lambda = 128$
Use binary tree as in Merkle, but...
• ...for security
  • pick index $i$ at random;
  • requires huge tree to avoid index collisions (e.g., height $h = 2\lambda = 256$).
• ...for efficiency:
  • use binary certification tree of OTS key pairs (= Hypertree with $d = h$),
  • all OTS secret keys are generated pseudorandomly.
SPHINCS \([\text{BHH}^+15]\)

- Select index pseudorandomly
- Use a few-time signature key-pair on leaves to sign messages
  - Few index collisions allowed
  - Allows to reduce tree height
- Use hypertree: Use \(d << h\).
SPHINCS$^+$ vs SPHINCS

• Allow for $2^{64}$ instead of $2^{50}$ signatures per key pair
• Add multi-target attack mitigation (Tweakable hash functions)
• “Simple” and “Robust” parameters
• New few-time signature scheme FORS
• Verifiable index selection
• Optional non-deterministic signatures
SPHINCS$^+$ in 3rd Round

Joint work with Jean-Philippe Aumasson, Daniel J. Bernstein, Ward Beullens, Christoph Dobr"{a}unig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan K"{o}lbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Bas Westerbaan

07 June 2021

https://sphincs.org/
3rd Round changes

- Two new team members: Ward Beullens, Bas Westerbaan
- New parameter sets (more efficient at same security)
- (Discussed hierarchical PRG & constant sum WOTS but discarded both)
New parameter sets

Search criteria:

• Improvement in optimized metric (fast / small)
• No significant penalty in other metric
• No worse verification speed
• No change to security assumptions / strength
• No increased complexity

-> We only changed h, d, log(t) & k
## New parameter sets

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>n</th>
<th>h</th>
<th>d</th>
<th>$\log(t)$</th>
<th>k</th>
<th>w</th>
<th>bitsec</th>
<th>sec level</th>
<th>sig bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPHINCS$^+$-128s</td>
<td>16</td>
<td>64</td>
<td>8</td>
<td>15</td>
<td>10</td>
<td>16</td>
<td>133</td>
<td>1</td>
<td>8 080</td>
</tr>
<tr>
<td>SPHINCS$^+$-128f</td>
<td>16</td>
<td>60</td>
<td>20</td>
<td>9</td>
<td>30</td>
<td>16</td>
<td>128</td>
<td>1</td>
<td>16 976</td>
</tr>
<tr>
<td>SPHINCS$^+$-192s</td>
<td>24</td>
<td>64</td>
<td>8</td>
<td>16</td>
<td>14</td>
<td>16</td>
<td>196</td>
<td>3</td>
<td>17 064</td>
</tr>
<tr>
<td>SPHINCS$^+$-192f</td>
<td>24</td>
<td>66</td>
<td>22</td>
<td>8</td>
<td>33</td>
<td>16</td>
<td>194</td>
<td>3</td>
<td>35 664</td>
</tr>
<tr>
<td>SPHINCS$^+$-256s</td>
<td>32</td>
<td>64</td>
<td>8</td>
<td>14</td>
<td>22</td>
<td>16</td>
<td>255</td>
<td>5</td>
<td>29 792</td>
</tr>
<tr>
<td>SPHINCS$^+$-256f</td>
<td>32</td>
<td>68</td>
<td>17</td>
<td>10</td>
<td>30</td>
<td>16</td>
<td>254</td>
<td>5</td>
<td>49 216</td>
</tr>
</tbody>
</table>
## New parameter sets

<table>
<thead>
<tr>
<th></th>
<th>n</th>
<th>h</th>
<th>d</th>
<th>log(t)</th>
<th>k</th>
<th>w</th>
<th>bitsec</th>
<th>sec level</th>
<th>sig bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPHINCS⁺-128s</td>
<td>16</td>
<td>64</td>
<td>8</td>
<td>15</td>
<td>10</td>
<td>16</td>
<td>133</td>
<td>1</td>
<td>8080</td>
</tr>
<tr>
<td>SPHINCS⁺-128f</td>
<td>16</td>
<td>60</td>
<td>20</td>
<td>9</td>
<td>30</td>
<td>16</td>
<td>128</td>
<td>1</td>
<td>16976</td>
</tr>
<tr>
<td>SPHINCS⁺-192s</td>
<td>24</td>
<td>64</td>
<td>8</td>
<td>16</td>
<td>14</td>
<td>16</td>
<td>196</td>
<td>3</td>
<td>17064</td>
</tr>
<tr>
<td>SPHINCS⁺-192f</td>
<td>24</td>
<td>66</td>
<td>22</td>
<td>8</td>
<td>33</td>
<td>16</td>
<td>194</td>
<td>3</td>
<td>35664</td>
</tr>
<tr>
<td>SPHINCS⁺-256s</td>
<td>32</td>
<td>64</td>
<td>8</td>
<td>14</td>
<td>22</td>
<td>16</td>
<td>255</td>
<td>5</td>
<td>29792</td>
</tr>
<tr>
<td>SPHINCS⁺-256f</td>
<td>32</td>
<td>68</td>
<td>17</td>
<td>10</td>
<td>30</td>
<td>16</td>
<td>254</td>
<td>5</td>
<td>49216</td>
</tr>
</tbody>
</table>
## New parameter sets

<table>
<thead>
<tr>
<th>sign</th>
<th>verify</th>
<th>sig</th>
<th>sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>128s</td>
<td>± 0</td>
<td>- 8 %</td>
<td>- 2.77 %</td>
</tr>
<tr>
<td>128f</td>
<td>- 24 %</td>
<td>+ 10 %</td>
<td>+ 0.66 %</td>
</tr>
<tr>
<td>192s</td>
<td>- 20 %</td>
<td>- 10 %</td>
<td>- 4.92 %</td>
</tr>
<tr>
<td>192f</td>
<td>± 0</td>
<td>± 0</td>
<td>± 0</td>
</tr>
<tr>
<td>256s</td>
<td>± 0</td>
<td>± 0</td>
<td>± 0</td>
</tr>
<tr>
<td>256f</td>
<td>- 13 %</td>
<td>± 0</td>
<td>+ 1.30 %</td>
</tr>
</tbody>
</table>

Changes in speed are averaged over robust / simple & SHA2, SHAKE & Haraka parameter sets. For more details see our change log and the latest specification.
After round three updates
H_msg with SHA-256 #1

Feb 11: Mail by Morgan Stern

“In particular, in SPHINCS+-SHA-256 there is an issue with the definition of the $H_{msg}$ function so that the security of the signature presently relies on the multi-target second pre-image resistance of the SHA-256 hash function.”

$$H_{msg}(R, PK.seed, PK.root, M) = \text{MGF1-SHA-256}(\text{SHA-256}(R||PK.seed||PK.root||M), m).$$
H_msg with SHA2-256 #1

- The multi-target second preimage attack loses about 64 bit in security
- Security down to 192 bits (for all SHA-256 parameters)
- Violates L5
- Fix: Switch to SHA2-512 for H_msg (& H_PRF) at L5.

$R | \ldots | || M$
Feb 16: Mail by John Kelsey

“I believe there’s also a long-message second preimage attack that applies here. (Ray Perlner pointed this out in a discussion.)”

Fix:

\[
H_{\text{msg}} : \quad = \text{MGF1−SHA−X}(R \parallel PK.\ seed \parallel \text{SHA−X}(R \parallel PK.\ seed \parallel PK.\ root \parallel M),m)
\]

(where X is 256 for L1 & L3, and 512 for L5)
H_msg with SHA2-256 #2

Fix:

\[
H_msg : \quad = MGF1-SHA-X(R || PK.seed || SHA-X(R || PK.seed || PK.root || M ), m)
\]

(where X is 256 for L1 & L3, and 512 for L5)

Attack:

1. Ask for q signatures on long messages (2^k message blocks)
2. Find expandable messages (takes time \(\sim O(2^{n/2})\))
3. Find collision between expandable message and a message block in long message (takes times \(O(2^{n-k-log q-1})\))
4. Expand expandable message sufficiently
H_msg with SHA2-256 #2

• Attack before fix takes time \( O(2^{n/2} + 2^{n-k-\log q-1}) \)

• Max values are \( q = 2^{64} \), \( k = 55 \) \( \Rightarrow \) We lose 119 bit security.

• Recall: Honest user signs!

• Assume compression function call takes \( 2^{-22} \) seconds (\( \approx 200ns \)).

• Attack takes \( 2^{64} \cdot 2^{55} = 2^{119} \) compression function calls.

• That is \( 2^{97} \) sec = \( 2^{72} \) years.

• Still \( 2^{52} \) years if key continuously used on 1 million machines!
Conclusion

• Possible synergies with standardizing stateful hash-based signatures
• *The* most conservative submission in the competition.
Thank you!

Questions?