HARAKA v2

Efficient Short-Input Hashing for Post-Quantum Applications

Stefan Kölbl\textsuperscript{1}  Martin M. Lauridsen\textsuperscript{2}  Florian Mendel\textsuperscript{3}  Christian Rechberger\textsuperscript{1,3}

March 7th, 2017

\textsuperscript{1}DTU Compute, Technical University of Denmark, Denmark

\textsuperscript{2}InfoSec Global Ltd., Switzerland

\textsuperscript{3}IAIK, Graz University of Technology, Austria
Impact of Quantum Computers

• Public-key
  • Diffie-Hellman
  • RSA
  • Elliptic Curves

• Symmetric-key
  • Block Ciphers
  • Hash Functions
Impact of Quantum Computers

• Public-key
  • Diffie-Hellman
  • RSA
  • Elliptic Curves

• Symmetric-key
  • Block Ciphers (Larger key)
  • Hash Functions (Longer output)
NIST-call\textsuperscript{1}

\begin{itemize}
  \item Digital Signature Scheme
  \item Encryption / Key Establishment
\end{itemize}

PQCrypto Project\textsuperscript{2}

\textsuperscript{1}\url{http://csrc.nist.gov/groups/ST/post-quantum-crypto/}
\textsuperscript{2}\url{https://pqcrypto.eu.org/}
Hash-based Signature Schemes

- Post-quantum secure
- Minimal Assumptions
- Lamport [Lam79], Merkle Tree [Mer89], XMSS [BDH11], SPHINCS [BHH+15], ...
Performance of hash-based signature schemes

- Many calls to the hash function...
- ...but using short input only.
- ...no collision resistance required.
Example SPHINCS:

- Provides 128-bit post-quantum security.
- Signing takes roughly 500,000 hash function evaluations.
Benchmarks from SUPERCOP on Intel Core i5-6600
Benchmarks from SUPERCOP on Intel Core i5-6600
A short-input hash function

- AES-based.
- 256- and 512-bit permutation.
- Using Davies-Meyer with 0 key.
Internal permutation of Haraka v2

- Substitution Permutation Network
- Round function: $\text{mix} \circ \text{aes}^m$
Haraka-256 v2

Requires only 6 instructions per round

- $4 \times \text{vaesenc}$
- $\text{vpunpckldq, vpunpckhdq}$
Haraka-512 v2

Requires only 16 instructions per round

- $8 \times \text{vaesenc}$
- 8 for $\text{mix}$
Security Analysis

• Active S-boxes
  • 80 for Haraka-256 v2
  • 130 for Haraka-512 v2

• Truncated Differentials

• Meet-in-the-Middle attacks

• Round Constants [Jea16]
Performance

- AES instructions have high latency.
- Costs for mixing can be hidden.
- Often multiple independent blocks available.
<table>
<thead>
<tr>
<th></th>
<th>Haswell</th>
<th></th>
<th>Skylake</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cycles/Byte</td>
<td>Cycles/Byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Haraka-256 v2</strong></td>
<td>1.25</td>
<td>0.72</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simpirav2([b = 2])</td>
<td>1.91</td>
<td>1.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SPHINCS-256-F</strong></td>
<td>11.31</td>
<td>11.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Haraka-512 v2</strong></td>
<td>1.75</td>
<td>0.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simpirav2([b = 4])</td>
<td>4.5</td>
<td>2.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SPHINCS-256-H</strong></td>
<td>11.16</td>
<td>10.92</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Multiple Inputs

<table>
<thead>
<tr>
<th></th>
<th>Haswell Cycles/Byte</th>
<th>Skylake Cycles/Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Haraka-256 v2</strong></td>
<td>1.14</td>
<td>0.63</td>
</tr>
<tr>
<td>Simpirav2[$b = 2$]</td>
<td>0.96</td>
<td>0.94</td>
</tr>
<tr>
<td>SPHINCS-256-$F$</td>
<td>2.11</td>
<td>1.71</td>
</tr>
<tr>
<td><strong>Haraka-512 v2</strong></td>
<td>1.43</td>
<td>0.72</td>
</tr>
<tr>
<td>Simpirav2[$b = 4$]</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>SPHINCS-256-$H$</td>
<td>1.99</td>
<td>1.62</td>
</tr>
</tbody>
</table>
# SPHINCS on Intel Skylake

<table>
<thead>
<tr>
<th></th>
<th>ChaCha12</th>
<th>Haraka v2&lt;sup&gt;3&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cycles</td>
<td>Cycles</td>
</tr>
<tr>
<td>Key generation</td>
<td>2,839,018</td>
<td>1,340,338 (×2.12)</td>
</tr>
<tr>
<td>Signing</td>
<td>43,517,538</td>
<td>20,782,894 (×2.09)</td>
</tr>
<tr>
<td>Verification</td>
<td>1,291,980</td>
<td>415,586 (×3.11)</td>
</tr>
</tbody>
</table>

<sup>3</sup>Updated numbers from [https://github.com/kste/haraka](https://github.com/kste/haraka).
Summary

• AES-based SPN for Short-Input Hash.
• Low Latency
• Can speed up SPHINCS significantly.

Future Work

• ARMv8 platform
• Collision vs. Preimage
CONCLUSION

Implementation of Haraka and SPHINCS-256-Haraka

https://github.com/kste/haraka
Questions?

