Update on the OPEN QUANTUM SAFE project

Christian Paquin

@chpaquin
Principal Program Manager
Microsoft Research

#QuantumSafeCryptography

ETSI/IQC Quantum Safe Cryptography - Technical Track
Nov 6-7, 2019
• Goal: support the development and prototyping of quantum-resistant cryptography

• Contributions from

• C library integrating multiple PQC algorithms under a common API
  • New v0.2 released in October 2019
  • New and updated (round 2) schemes

• Integrations into boringssl, OpenSSL, OpenSSH, OpenVPN (MSR)
  • New: PQ/hybrid KEX/auth in TLS 1.3 and SSH

• C++, C#, Go, and Python wrappers

• Instructions to build nginx, apache, chromium

https://openquantumsafe.org/
Prototyping PQC paper

- Analyze various options to integrate PQC into TLS and SSH
- Focus on hybrid scenarios
- Lessons learned from OpenSSL, OpenSSH, and s2n integrations

https://eprint.iacr.org/2019/858
Hybrid scenarios

- Early migration should use a hybrid of classical/PQ schemes
  - Security of today + safety net against quantum computer
  - Secure if one of the two is secure
- TLS and SSH negotiate algorithms, but not two at the same time. We need to define either:
  - new combo schemes, e.g. ECDHE-SIKEp503:
    - Easy to implement, backward compatible
  - a new hybrid approach:
    - Flexible negotiation (algs selected separately), need spec/code changes
- Consider backward compatibility, performance, latency, data flow
- Implemented approach: combo schemes and concatenation of keys, ciphertexts, and signatures

ETSI/IQC QSC 2019 – Christian Paquin (MSR)
TLS case study

- Added PQ/hybrid KEX & auth
- TLS 1.2 (OpenSSL 1.0.2)
  - Explosion of schemes (specifies KEX, auth, symmetric cipher, hash)
  - Spec message size limit: $2^{24}$ bytes. OpenSSL limit smaller
  - Tested with OpenSSL tools, apache, OpenVPN
- TLS 1.3 (OpenSSL 1.1.1)
  - PQ algs masquerade as EC curves
  - Concat strategy more secure than 1.2 (KDF hashes transcripts)
  - Spec pub key and sig limit: $2^{16} - 1$ bytes, cert limit: $2^{24} - 1$ bytes. OpenSSL limit is smaller
  - Tested with OpenSSL tools, nginx
- [https://github.com/open-quantum-safe/openssl](https://github.com/open-quantum-safe/openssl)
SSH case study

- Added PQ/hybrid KEX & auth to OpenSSH
- Define new algorithms, e.g.: ecdh-nistp384-sike-503-sha384@openquantumsafe.org
- Supports both client and server public key authentication
- Spec message size limit: $2^{32}$ bytes, large enough for all round 2 candidates, but OpenSSH limit is smaller ($2^{18}$)
- https://github.com/open-quantum-safe/openssh-portable
# Key Encapsulation Mechanisms

<table>
<thead>
<tr>
<th>KEM scheme</th>
<th>OpenSSL 1.0.2 TLS 1.2</th>
<th>OpenSSL 1.1.1 TLS 1.3</th>
<th>OpenSSH 7.9 SSH2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIKE 1/2/3 L1/3/5 (round 1)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Frodo KEM 640/976 AES/SHAKE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Frodo KEM 1344 AES/SHAKE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Kyber 512/768/1024</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LEDAcrypt KEM LT 12/32/52</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NewHope 512/1024 CCA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NTRU HPS (2048-509/677)/(4096-821)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NTRU HRSS 701</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NTS KEM (12,64)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>LightSaber/Saber/FireSaber KEM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SIKE p434/p503/p610/p751</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

KEM integrations for both PQ and hybrid (with ECDHE)

**Legend:**
- ✓ Success
- ✓ Works with code mods
- ✗ Did not work

ETS/ICT QSC 2019 – Christian Paquin (MSR)
## Signatures

<table>
<thead>
<tr>
<th>KEM scheme</th>
<th>OpenSSL 1.1.1 TLS 1.3</th>
<th>OpenSSH 7.9 SSH2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dilithium 2/3/4</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MQDSS 31 48/64</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Picnic L1 FS/UR</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Picnic L3/L5 FS/UR</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Picnic2 L1 FS</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Picnic2 L3/L5 FS</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>qTesla I/III-size/III-speed (round 1)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rainbow Ia Classic</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rainbow Ia Cyclic/Compressed</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rainbow IIIc/Vc Classic/Cyclic/Compressed</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>SPHINCS+ * 128s *</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SPHINCS+ * 128f/192f/192s/256f/256s *</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Signature integrations for both PQ and hybrid (with ECDSA)

Legend:
- ✓ Success
- ✓ Works with code mods
- x Did not work

ETS/IQC QSC 2019 – Christian Paquin (MSR)
What’s next?

For us
• Test all round 2 schemes
• Performance test
• More protocols

For you
• Start planning migration to PQC
• Start using some tools (SSH, OpenVPN)

cpaquin@microsoft.com
@chpaquin