Updates from the Open Quantum Safe project

John Schanck

OQS core team: Michael Baentsch, Eric Crockett (AWS), Vlad Gheorghiu (University of Waterloo), Basil Hess (IBM Research), Christian Paquin (Microsoft Research), John Schanck (University of Waterloo), Douglas Stebila (University of Waterloo), Goutam Tamvada (University of Waterloo)

https://openquantumsafe.org
https://github.com/open-quantum-safe
Open Quantum Safe Project

Use in applications
- Apache
- httpd
- nginx
- curl, links
- OpenVPN
- Chromium
- OpenSSL
  S/MIME, TLS 1.3, X.509
  OpenSSL 3 provider
- BoringSSL
- Open SSH
- Language SDKs
  C#, C++, Go, Java, Python, Rust

Integration into forks of widely used open-source projects
- C language library, common API
  - x86/x64 (Linux, Mac, Windows)
  - ARM (Android, Linux)

Integration into forks of widely used open-source projects
- Chromium

C language library, common API
- x86/x64 (Linux, Mac, Windows)
- ARM (Android, Linux)

Key exchange / KEMs
- isogenies
- code-based
- lattice-based

Signatures
- multi-variate polynomial
- hash-based / symmetric

Industry partners:
- Amazon Web Services
- evolutionQ
- IBM Research
- Microsoft Research

Additional contributors:
- Cisco
- Senetas
- PQClean project
- Individuals

Financial support:
- AWS
- Canadian Centre for Cyber Security
- NSERC
- Unitary Fund

liboqs

- Implementations from PQClean or direct contribution
- MIT License (and other free licenses)
- Builds on Windows, macOS, Linux; x86-64, ARM32v7, ARM64v8
- Wrappers for C++, Go, Java, .Net, Python, Rust

https://openquantumsafe.org/liboqs/
liboqs

• Version 0.5.0 released March 2021
  • Includes all Round 3 submissions (except GeMSS)
    • Some implementations still Round 2 versions
  • More robust testing:
    • LLVM address and undefined behavior sanitizers
    • Secret-dependent branching using Valgrind

• Version 0.6.0 to be released in June 2021
  • Algorithm updates
  • Common code deduplication (SHA3)
  • New build options and cross-compilation support
  • Improved code dispatching

https://openquantumsafe.org/liboqs/
# TLS 1.3 implementations

<table>
<thead>
<tr>
<th>Feature</th>
<th>OQS-OpenSSL 1.1.1</th>
<th>OQS-OpenSSL 3 provider</th>
<th>OQS-BoringSSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PQ key exchange in TLS 1.3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Hybrid key exchange in TLS 1.3</td>
<td>Yes</td>
<td>Coming soon</td>
<td>Yes</td>
</tr>
<tr>
<td>PQ certificates and signature authentication in TLS 1.3</td>
<td>Yes</td>
<td>API change required</td>
<td>Yes</td>
</tr>
<tr>
<td>Hybrid certificates and signature authentication in TLS 1.3</td>
<td>Yes</td>
<td>API change required</td>
<td>No</td>
</tr>
</tbody>
</table>

Using draft-ietf-tls-hybrid-design for hybrid key exchange

Interoperability test server running at [https://test.openquantumsafe.org](https://test.openquantumsafe.org)

[https://openquantumsafe.org/applications/tls/](https://openquantumsafe.org/applications/tls/)
Applications

- Demo integrations into:
  - Apache
  - nginx
  - haproxy
  - curl
  - Chromium
- Docker images available.

- In most cases integration of updated OpenSSL required few/no modifications to application.
- Some algorithm-specific issues remain.

https://openquantumsafe.org/applications/tls/#demo-integrations
Other protocols

**SSH**
- Fork of OpenSSH v7 (soon: v8)
- PQ and hybrid key exchange
- PQ and hybrid authentication

**CMS/SMIME**
- In fork of OpenSSL
- PQ and hybrid signatures

**X.509**
- In fork of OpenSSL
- PQ and hybrid signatures
Benchmarking

https://openquantumsafe.org/benchmarking/

• Core algorithm speed and memory usage

• TLS performance in ideal network conditions

• Currently benchmarking on:
  • Intel Cannon Lake
  • ARM Cortex-A72 (reference code only)
Use in prototyping & research

- **Cisco**: Post-quantum TLS 1.3 and SSH performance (preliminary results)

- **IBM**: IBM Cloud delivers quantum-safe cryptography and Hyper Protect Crypto Services to help protect data in the hybrid era
  - [https://github.com/IBM/qsc-ingress](https://github.com/IBM/qsc-ingress)

- **Microsoft Research**: Post-quantum cryptography VPN
  - [https://github.com/Microsoft/PQCrypto-VPN](https://github.com/Microsoft/PQCrypto-VPN)

- **strongSwan**: Post-quantum cryptography in IKEv2 using strongSwan
  - [https://github.com/strongX509/docker/tree/master/pq-strongswan](https://github.com/strongX509/docker/tree/master/pq-strongswan)


- **Towards quantum-safe VPNs and Internet**, by Maran van Heesch, Niels van Adrichem, Thomas Attema, and Thijs Veugen.


[https://openquantumsafe.org/research/](https://openquantumsafe.org/research/)
Contributions welcome!

https://github.com/open-quantum-safe/