

# OPEN QUANTUM SAFE

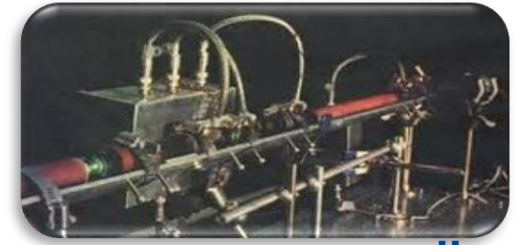


Deployment of PQC  
Institut Henri Poincaré  
October 7 to 11, 2024

Christian Paquin



# About me



Université   
de Montréal

- Studied quantum cryptography 25+ years ago at University of Montreal
- Worked in the industry as a cryptographic engineer
- Now with the *MSR Cryptography* team, working on
  - Post-quantum cryptography
  - Privacy-preserving identity (anonymous credentials, ZK proofs)
  - Content provenance and authenticity (C2PA)
- Links
  - MSR page: <https://www.microsoft.com/en-us/research/people/cpaquin/>
  - Blog: <https://christianpaquin.github.io/>

# OPEN QUANTUM SAFE

- Development and prototyping of quantum-resistant cryptography
- *liboqs*: C library offering PQC algorithms
- Bindings for [C++](#), [C#](#), [go](#), [java](#), [python](#), [rust](#)
- Protocol integration into TLS and CMS (OpenSSL, BoringSSL), SSH (OpenSSH, libssh)
- Application integration into curl, chromium, httpd, nginx, openvpn, quic, wireshark, and more
- Supports pure PQC and hybrid modes
- Now part of the Linux Foundation PQC Alliance

<https://openquantumsafe.org>



Financial and in-kind support:



CANADIAN CENTRE FOR CYBER SECURITY | CENTRE CANADIEN DE CYBERSÉCURITÉ



IBM Research



# Historical timeline

## Post-quantum key exchange for the TLS protocol from the ring learning with errors problem

Joppe W. Bos<sup>1</sup>, Craig Costello<sup>2</sup>, Michael Naehrig<sup>2</sup>, and Douglas Stebila<sup>3,\*</sup>

<sup>1</sup> NXP Semiconductors, Leuven, Belgium

<sup>2</sup> Microsoft Research, Redmond, Washington, USA

<sup>3</sup> University of Waterloo, Canada



## PQC Call for proposals

### 2015

- BCNF paper (May)

### 2016

- Initial OQS commit (Aug)
- Windows support (Sept)
- SIDH (Nov)

### 2017

- NIST Round 1 [69 algs] (Dec)
- Sig API – Picnic (Dec)

### 2018

- OpenSSL 1.1.1 TLS 1.3 (Apr)
- TLS 1.3 Auth (Jun)
- OQS v0.1.0 (Dec)

### 2019

- NIST Round 2 [26 algs] (Jan)
- OQS v0.2.0 (Oct)

### 2020

- NIST Round 3
- OQS v0.3 & v0.4

### 2021

- OQS v0.5 - v0.7
- OpenSSL 3.0 Provider (Feb)

### 2022

- 4 selected algs
- NIST Round 4
- OQS v0.7.2

### 2023

- Signature RFP (Jun)
- Draft standards (Aug)
- OQS v0.8 - v0.9

### 2024

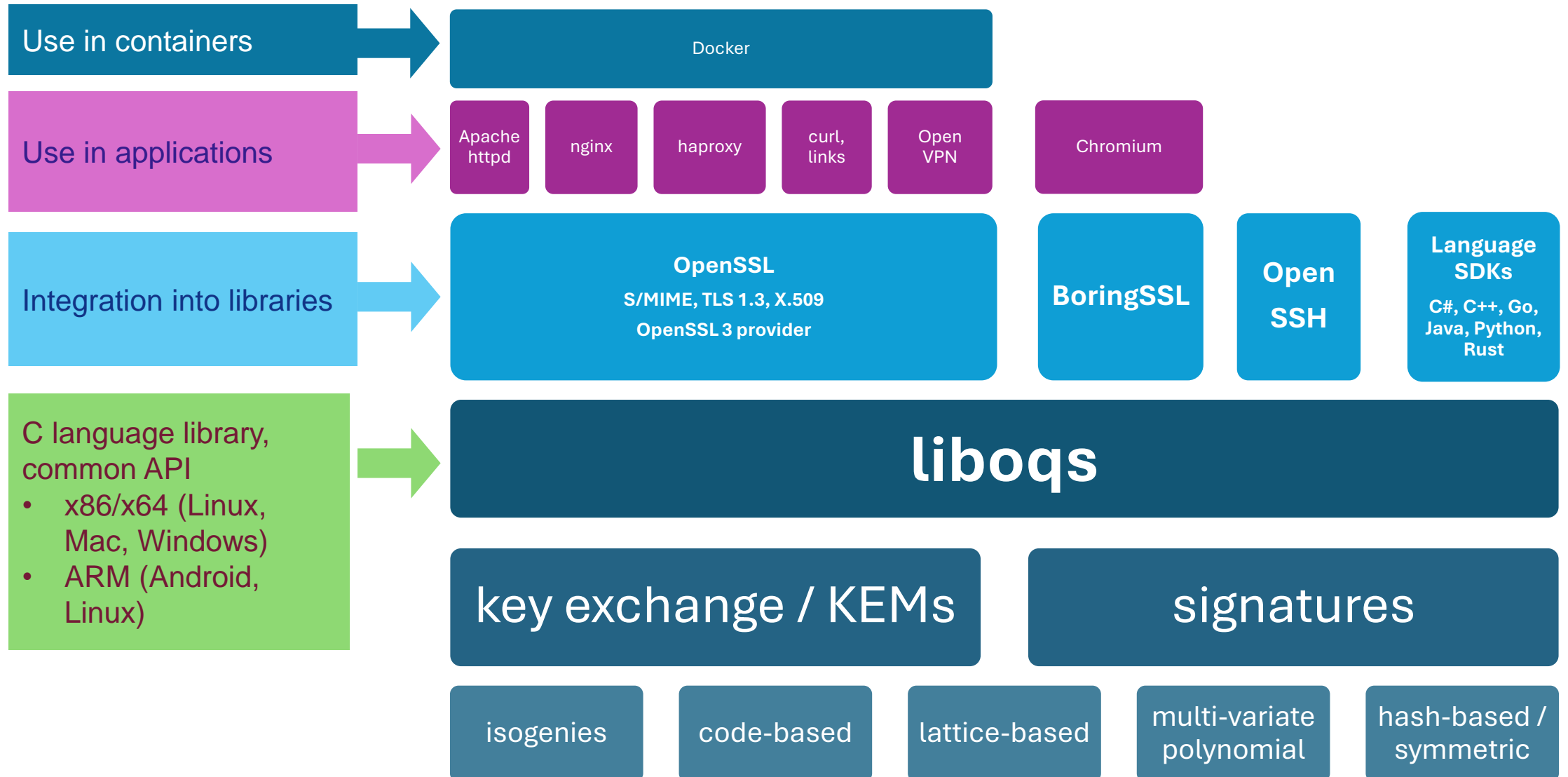
- Linux Foundation (Jan)
- FIPS Released (Aug)
- OQS v0.10 - v0.11



## Post-Quantum Cryptography Alliance

- PQCA launched in February 2024
- Projects
  - Open Quantum Safe
    - Support the development and prototyping of quantum-resistant cryptography
  - PQ Code Package
    - High-assurance implementations of standards-track post-quantum cryptography algorithms
- Goal: provide support to make OQS more robust and market ready

# OQS Architecture



# Latest release: liboqs 0.11.0

- Released Sept 27, 2024
- Updates
  - ML-KEM support
  - Adds MAYO, CROSS signatures from Round 1 Additional Signatures
  - Adds XMSS, LMS (disabled by default)
  - Verified implementations of Kyber-512, -768 from libjade
- Supported algorithms
  - KEM: BIKE, Classic McEliece, FrodoKEM, HQC, Kyber, ML-KEM, NTRU-Prime
  - Sig: CROSS, Dilithium, Falcon, MAYO, ML-DSA (IPD), SPHINCS+, XMSS, LMS

<https://github.com/open-quantum-safe/liboqs/releases/tag/0.11.0>

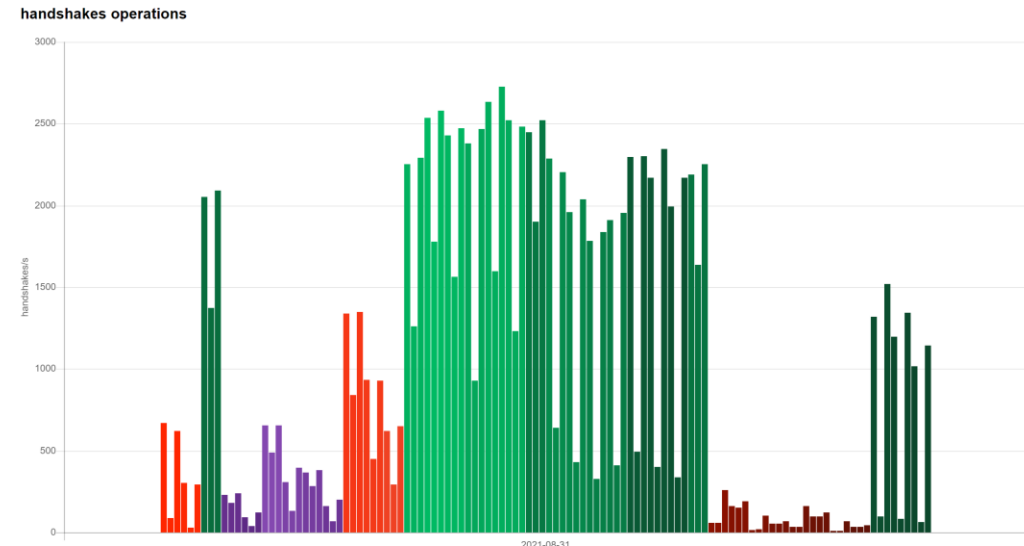
# TLS

- Specification
  - Hybrid KEM and sig: <https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/>
  - X.509: <https://datatracker.ietf.org/doc/draft-ietf-lamps-dilithium-certificates/>
- OpenSSL 3.0 Provider
  - <https://github.com/open-quantum-safe/oqs-provider/>
  - The recommended way to integrate PQC into OpenSSL 3.0
  - Upcoming integration into OpenSSL announced:  
<https://openssl-library.org/post/2024-09-17-post-quantum/>
- Experimental BoringSSL support
  - <https://github.com/open-quantum-safe/boringssl>
- Interoperability server: <https://test.openquantumsafe.org/>



# TLS Benchmarking

- Core algorithm speed and memory usage
- TLS performance (PQ-only & hybrid)
- Support Intel AVX2 and ARM 64
- <https://openquantumsafe.org/benchmarking/>



## KEM performance

Operations per second per algorithm

Raw Data	2021-08-31					
Algorithm	keygen/s	keygen(cycles)	encaps/s	encaps(cycles)	decaps/s	decaps(cycles)
FrodoKEM-1344-AES (x86_64)	588.67	4247027	461.18	5421019	478.51	5223477
FrodoKEM-1344-AES (x86_64-ref)	555.00	4504727	352.55	7090322	358.21	6979490
FrodoKEM-1344-AES (x86_64-noport)	612.13	4083887	481.00	5197275	496.17	5038778
FrodoKEM-1344-SHAKE (x86_64)	207.39	12055505	192.01	13017521	193.08	12949723
FrodoKEM-1344-SHAKE (x86_64-ref)	86.91	28760679	78.46	31867850	72.69	34393744
FrodoKEM-1344-SHAKE (x86_64-noport)	252.25	9910764	240.25	10404343	243.34	10273587
FrodoKEM-640-AES (x86_64)	2184.33	1144361	1503.00	1663334	1653.00	1512282
FrodoKEM-640-AES (x86_64-ref)	1971.67	1267907	1254.67	1992566	1308.33	1910734
FrodoKEM-640-AES (x86_64-noport)	2389.33	1046066	1719.67	1453571	1801.33	1387905
FrodoKEM-640-SHAKE (x86_64)	793.67	3149835	700.67	3567992	717.43	3484136
FrodoKEM-640-SHAKE (x86_64-ref)	350.88	7124020	320.12	7808764	323.45	7728289
FrodoKEM-640-SHAKE (x86_64-noport)	1042.67	2397485	892.00	2802818	863.67	2894581
FrodoKEM-976-AES (x86_64)	1034.32	2416590	781.67	3198448	816.39	3061548
FrodoKEM-976-AES (x86_64-ref)	927.36	2695800	601.13	4158205	639.33	3910549
FrodoKEM-976-AES (x86_64-noport)	1098.97	2274447	841.00	2972547	879.33	2842783
FrodoKEM-976-SHAKE (x86_64)	370.88	6740124	339.44	7364107	315.56	7921259
FrodoKEM-976-SHAKE (x86_64-ref)	155.56	16070978	143.05	17476562	143.90	17369605
FrodoKEM-976-SHAKE (x86_64-noport)	478.51	5223916	430.05	5812636	441.00	5669368

## SIG memory consumption

Bytes per algorithm

Raw Data		2021-08-31				
Algorithm	keygen(maxHeap)	keygen(maxStack)	sign(maxHeap)	sign(maxStack)	verify(maxHeap)	verify(maxStack)
Dilithium2 (x86_64)	12656	20880	11584	49616	11360	21592
Dilithium2 (x86_64-ref)	12656	38952	15400	51360	10752	37120
Dilithium2-AES (x86_64)	12656	17040	15400	44816	10528	14800
Dilithium2-AES (x86_64-ref)	12656	38952	15400	51360	10752	37816
Falcon-512 (x86_64)	10994	19216	11784	40944	11784	2320
Falcon-512 (x86_64-ref)	10994	19200	11784	40944	11784	2304

# SSH

- Implements [draft-kampanakis-curdle-ssh-pq-ke-04](#)
- OpenSSH: OQS enabled version of OpenSSH v9.7
  - KEM: BIKE, ClassicMcEliece, FrodoKEM, HQC, Kyber, ML-KEM, NTRU-Prime
  - Sig: Dilithium, Falcon, MAYO, ML-DSA, SPHINCS+
- Libssh (inactive)

# OQS Demos

```
cpaquin@CPAQUINSB3: ~/dev X cpaquin@CPAQUINSB3: ~/dev X + v - □ X
(base) cpaquin@CPAQUINSB3:~/dev/oqs/LibOqs$ docker run --network nginx-test --name oqs-nginx
-p 4433:4433 openquantumsafe/nginx

172.18.0.3 - - [04/Oct/2024:20:23:27 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.81.0"
-
```

```
cpaquin@CPAQUINSB3: ~/dev X cpaquin@CPAQUINSB3: ~/dev X + v - □ X
(base) cpaquin@CPAQUINSB3:~/dev/oqs/LibOqs$ docker run --network nginx-test -it openquantumsafe
/curl curl -k https://oqs-nginx:4433 --curves p384_kyber768
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
(base) cpaquin@CPAQUINSB3:~/dev/oqs/LibOqs$
```



## Supported

- curl
- Apache httpd
- nginx
- Chromium

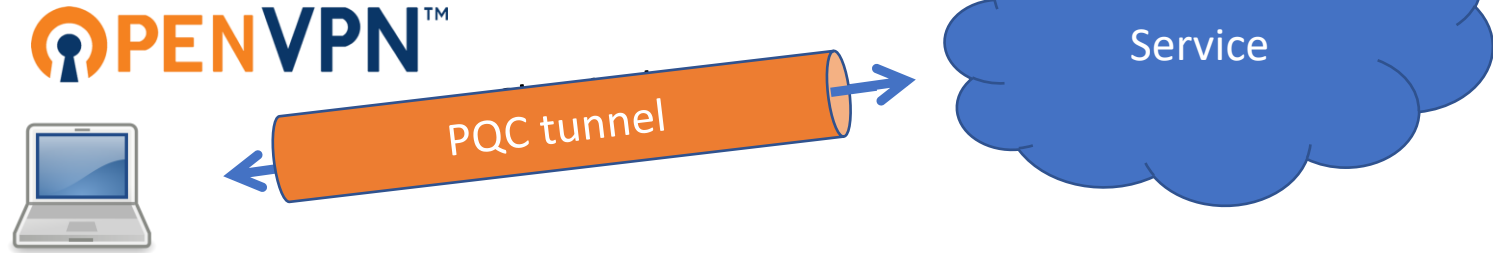
## Unsupported

- OpenSSH
- Wireshark
- Epiphany
- OpenVPN
- ngtcp2
- OpenLiteSpeed
- h2load
- HAproxy
- Mosquitto
- Envoy
- Unbound

<https://github.com/open-quantum-safe/oqs-demos>

# PQ VPN tunnels

- OpenVPN integration
  - Uses OQS's OpenSSL fork
  - Easy legacy app tunneling
  - <https://www.microsoft.com/en-us/research/project/post-quantum-crypto-vpn/>



- Project Natick PQC VPN experiment
  - Natick was an underwater datacenter module off the coast of Scotland
  - We ran a PQ VPN from Redmond
    - Used ECDHE-P256 + SIKEp434 hybrid
  - <https://www.microsoft.com/en-us/research/project/post-quantum-crypto-tunnel-to-the-underwater-datacenter/>








# Migration to PQC project

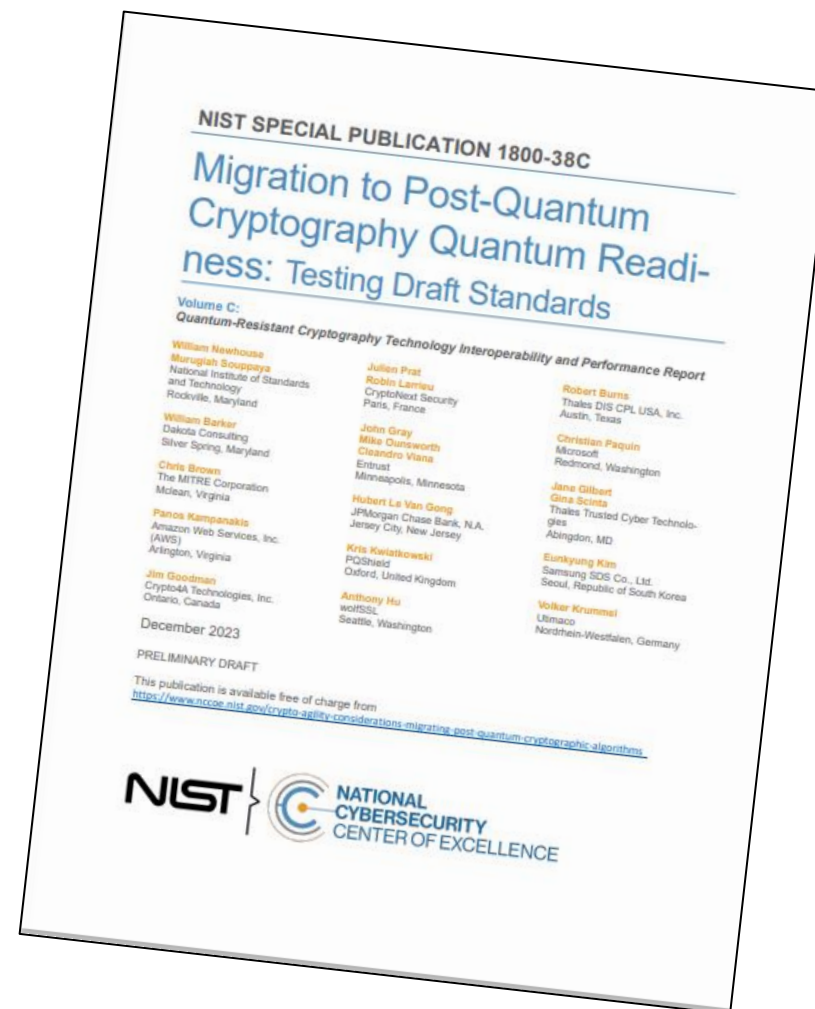
## Two workstreams

- Vulnerable cryptography detection
- Interoperability and performance

- [Amazon Web Services, Inc. \(AWS\)](#)
- [ATIS](#)
- [Cisco Systems, Inc.](#)
- [Comcast](#)
- [Crypto4A Technologies, Inc.](#)
- [CryptoNext Security](#)
- [Cybersecurity and Infrastructure Security Agency \(CISA\)](#)
- [Data-Warehouse GbmH](#)
- [Dell Technologies](#)
- [DigiCert](#)
- [Entrust](#)
- [Gutsy](#)
- [HP, Inc.](#)
- [HSBC](#)
- [IDEMIA Secure Transactions](#)
- [IBM](#)
- [Information Security Corporation](#)
- [InfoSec Global](#)
- [ISARA Corporation](#)
- [JPMorgan Chase Bank, N.A.](#)
- [Keyfactor](#)
- [Kudelski IoT](#)
- [Microsoft](#)
- [National Security Agency \(NSA\)](#)
- [NXP Semiconductors](#)
- [Palo Alto Networks](#)
- [Post-Quantum](#)
- [PQShield](#)
- [QuantumXchange](#)
- [SafeLogic, Inc.](#)
- [Samsung SDS Co., Ltd.](#)
- [SandboxAQ](#)
- [Santander](#)
- [SSH Communications Security Corp](#)
- [Thales DIS CPL USA, Inc.](#)
- [Thales Trusted Cyber Technologies](#)
- [Utimaco](#)
- [Verizon](#)
- [wolfSSL](#)

# NCCoE – Interoperability & Performance

- Testing PQC integration in
  - TLS 
  - SSH 
  - HSM
  - X.509 
  - VPN







## Get involved

OQS has been at the forefront of PQC standardization, prototyping, and product integration.

The work continues!



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



<https://discord.gg/qRfMantKwc>

cpaquin@microsoft.com