OPEN QUANTUM SAFE

software for the transition to quantum-resistant cryptography

Douglas Stebila, University of Waterloo

https://openquantumsafe.org/ • https://github.com/open-quantum-safe/ • https://www.douglas.stebila.ca/research/presentations/

My history with OpenSSL

- OpenSSL was the first open source software project I contributed to
- First commit: 2002, included in OpenSSL 0.9.8

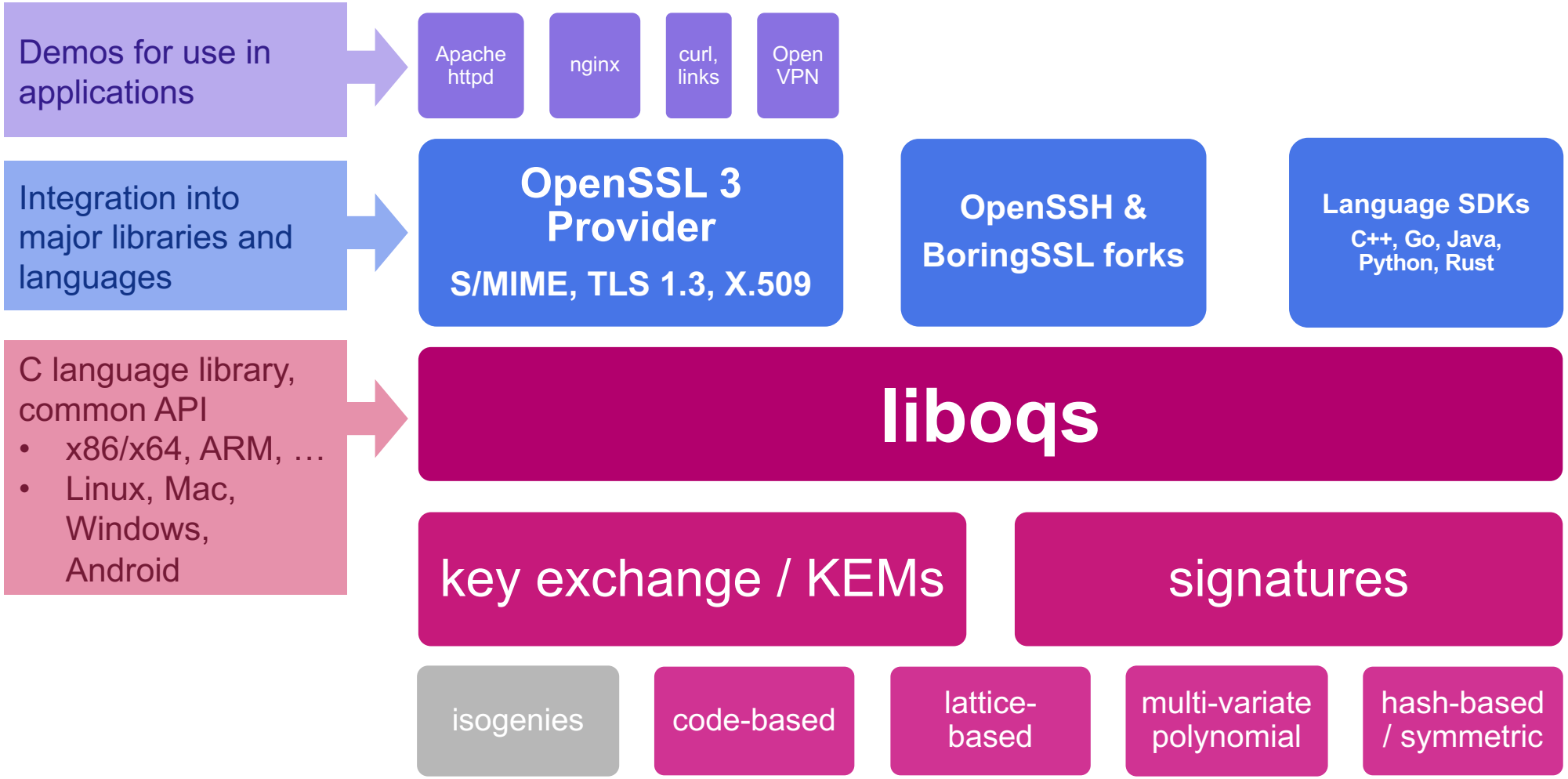
- Intern at Sun Microsystems
- Project: evaluate suitability of elliptic curve cryptography for adoption and build prototypes
  - Paper on benchmarking elliptic curve cryptography in SSL/TLS based on OpenSSL integration

My history with OpenSSL lead to OQS

- In 2014, I started a paper [BCNS15] with colleagues about key exchange based on the ring learning with errors problem
- Goal: embed RLWE into key exchange, pick parameters, implement, evaluate in TLS
- Time to revisit my work from a decade ago about adding a new algorithm to OpenSSL and benchmarking it
- I decided to post my OpenSSL 1.0.1 fork containing RLWE key exchange on Github
- This was the seed of the Open Quantum Safe project, which started in 2016

[BCNS15] Bos, Costello, Naehrig, Stebila. IEEE S&P 2015.

Open Quantum Safe Project



https://openquantumsafe.org/ • https://github.com/open-quantum-safe/

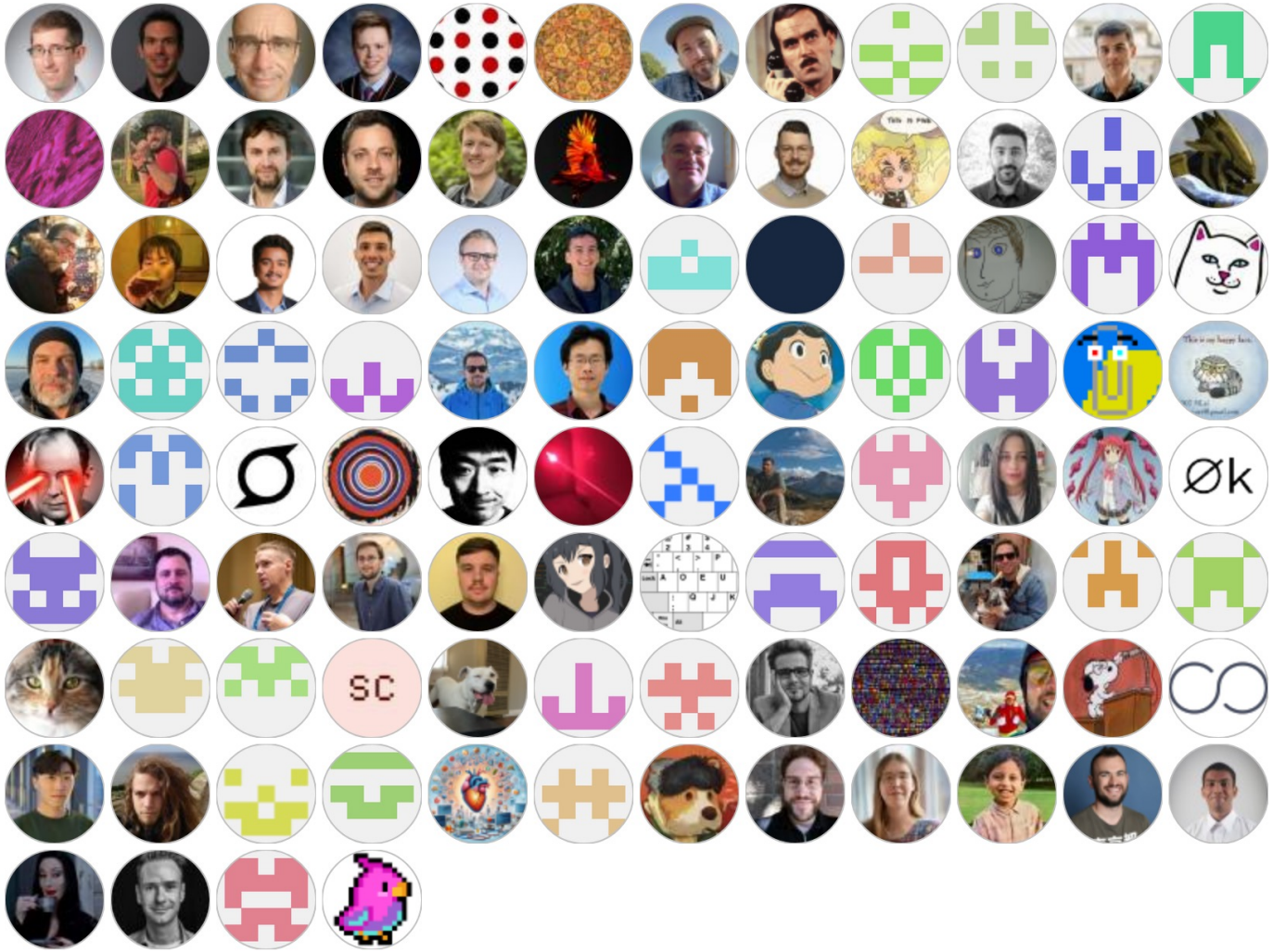
liboqs

C library with common API for post-quantum signature schemes and key encapsulation mechanisms

- MIT License and others
- Builds on Windows, macOS, Linux; x86_64, ARM v8, …

https://openquantumsafe.org/liboqs/ • https://github.com/open-quantum-safe/liboqs/blob/main/PLATFORMS.md

liboqs contributors



https://openquantumsafe.org/team/

Algorithm families

18 current algorithm families

NIST standards	ML-KEM, ML-DSA, SLH-DSA
NIST + IRTF standards	LMS, XMSS
NIST selections	Falcon, HQC
ISO draft standards	Classic McEliece, FrodoKEM
NIST signature on-ramp	CROSS, MAYO, SNOVA, UOV
Other	BIKE, Kyber Round 3, NTRU, NTRU-Prime, SPHINCS+

In total: 326 algorithm variants currently available

https://openquantumsafe.org/liboqs/algorithms/

Current status of liboqs

- Constant-time testing of most algorithms (using Valgrind)
- Passing ACVP tests for ML-KEM and ML-DSA
  - Some third-parties have CAVP certificates for ML-KEM in liboqs

Future directions for liboqs

Primary focus continues to be on supporting research and experimentation.

No plans for FIPS validation by the OQS team.

Algorithm updates

- Update standards-track algorithms: Falcon, HQC, FrodoKEM
- Support more ML-DSA options (externalMu, prehash, deterministic)
- Integrate formally verified code

https://github.com/open-quantum-safe/liboqs/wiki/Contribution-wishlist

Future directions for liboqs

Signature on-ramp

- Solicit more schemes from NIST signature on-ramp round 2

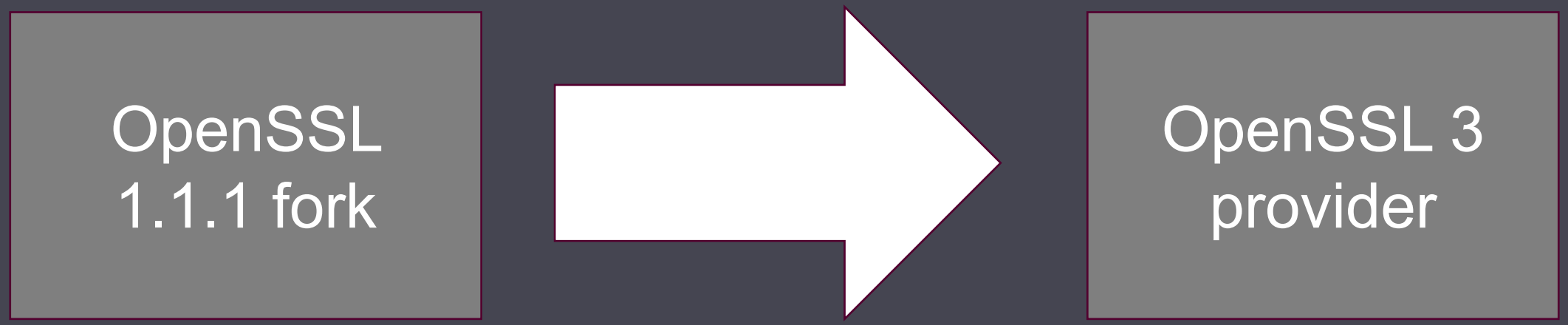
Looking to grow the contributor community to help achieve these and other future directions.

Improved testing

- Constant-time testing on more platforms
- Explore data-independent timing features of CPUs

https://github.com/open-quantum-safe/liboqs/wiki/Contribution-wishlist

Integration into TLS



OQS Provider for OpenSSL 3+

Adds support for all PQ algorithms in liboqs to OpenSSL 3+ using the OpenSSL provider architecture

Functionality progressively available based on OpenSSL 3.x version

- Hybrid + pure key exchange in TLS 1.3
- Hybrid + pure authentication in TLS 1.3
- Hybrid + pure signatures in CMS, CMP, X.509, and related formats

Talk to Michael Baentsch at the OpenSSL conference!

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

OpenSSL and OQS

Great to have NIST PQ standards in OpenSSL 3.5!

- Looking forward to see continued growth of PQ within OpenSSL

OQS Provider brings PQ algorithm support to users still relying on OpenSSL 3.0–3.4.

OQS Provider brings many experimental PQ algorithms for testing and evaluation.

Future directions for OQS Provider

- Seeking participation from the community
- Community members are interested in:
  - Seed support
  - Ability to use implementations from a provider like OQS even when algorithm is supported in OpenSSL 3.5

OQS Provider community call on Thursday October 23 at 15:30 UTC
See https://pqca.org/calendar/

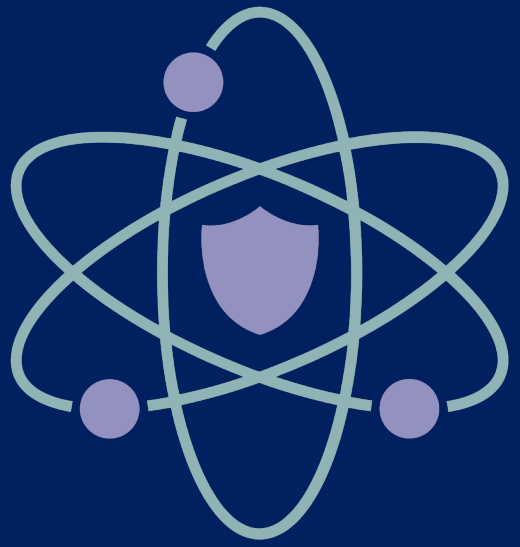
https://github.com/open-quantum-safe/oqs-provider/discussions/701

Getting involved in OQS

- Would love to have your contributions of ideas and code
- Tell us what features you would like from OQS

- OQS Provider community call: Thursday October 23 at 15:30 UTC https://pqca.org/calendar/
- Talk to Norm Ashley and Michael Baentsch at the OpenSSL Conference this week!
- Visit us on Github https://github.com/open-quantum-safe
- Pop in to one of our weekly calls

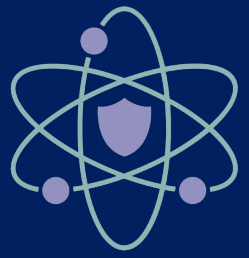
Not-for-profit organization hosting many open source projects (including e.g. Linux kernel and Kubernetes)



Post-Quantum Cryptography Alliance

To advance the adoption of post-quantum cryptography, by producing high-assurance software implementations of standardized algorithms, and supporting the continued development and standardization of new post-quantum algorithms with software for evaluation and prototyping.

https://pqca.org • https://github.com/PQCA



Post-Quantum Cryptography Alliance

Projects

Open Quantum Safe

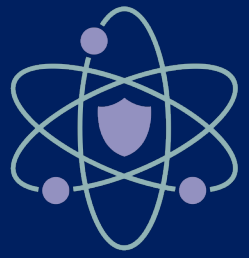
Support the development and prototyping of quantum-resistant cryptography

- **liboqs**: Library of standards-track post-quantum digital signature and public key encryption schemes
- **OQS Provider**: Integration of PQ algorithms into OpenSSL 3

PQ Code Package

Creating and maintaining high-assurance implementations of standards-track post-quantum cryptography algorithms

- mlkem-native
- mldsa-native, slhdsa-native, and more



Post-Quantum Cryptography Alliance

Getting involved

- All development done under open source licenses (MIT, Apache 2)
- Participation open to all
- Organizations can join as members to influence budget and direction

https://pqca.org • https://github.com/PQCA

PQ Code Package

mlkem-native, mldsa-native, and more

Hanno Becker (AWS), Matthias J. Kannwischer (Chelpis Quantum Corp.)
and the PQ Code Package team

https://github.com/pq-code-package

PQ Code Package

Mission

High-speed, high-assurance implementations of NIST-standardized post-quantum cryptographic algorithms

Focus

Production-ready code with rigorous security analysis

- Constant time
- High-assurance implementations
- Multiple languages, platforms
- Liberally licensed: Default is Apache 2.0 (mlkem-native/mldsa-native: Apache-2.0 OR ISC OR MIT)

https://github.com/pq-code-package/

PQ Code Package • Current projects

mlkem-native

mldsa-native

slhdsa-c

mlkem-libjade

mlkem-rust-libcrux

mlkem-native

- A secure, fast, and portable C90 implementation of ML-KEM
  - Fork of the ML-KEM reference implementation

- C code is proved memory-safe (no memory overflow) and type-safe (no integer overflow) using CBMC
- AArch64 assembly is proved functionally correct at the object code level using HOL-Light
- Constant-time testing for absence of secret-dependent branches, memory-access patterns and variable-latency instructions via Valgrind

- Used in AWS' Cryptography library AWS-LC; rustls; liboqs
- Chelpis has CAVP certificate #39634 for mlkem-native

https://github.com/pq-code-package/mlkem-native

Formal verification in mlkem-native

C code

Goal: Type-safety and memory-safety (no overflows)

Tool: C Bounded Model Checker (CBMC)

Approach: Automatic proofs from per-function in-source contracts, invariants, bounds

Coverage: All C code

Continuous Integration: Runs on every change (~15 min/parameter set)

Assembly

Goal: Prove assembly object code implements mathematical spec

Tool: HOL-Light: Interactive theorem prover by John Harrison

Approach: Manual proof per assembly function (using s2n-bignum infrastructure for e.g., instruction semantics)

Coverage: All AArch64 assembly (x86_64 next) – largely contributed by John Harrison

Continuous Integration: Runs on every change (up to 2 hours/function)

https://github.com/diffblue/cbmc • https://github.com/jrh13/hol-light

Future directions for PQ Code Package

mlkem-native

- New backends:
  - 64-bit RISC-V
  - PowerPC ppc64le
  - M-profile Vector Extension / Helium

mldsa-native

- Work in progress
- Same goals as mlkem-native
- Working to close performance gaps
- Alpha release in the next few months

Open Quantum Safe & friends

Douglas Stebila, University of Waterloo

Open Quantum Safe

- liboqs: standards-track + experimental PQ algorithms
- OQS Provider: integration into OpenSSL 3+
- Get involved!
  - OQS Provider Community Call Thursday October 23

PQ Code Package

- mlkem-native: secure, fast, portable C and assembly
- mldsa-native: work-in-progress
- Get involved!

Post-Quantum Cryptography Alliance