New Initiatives in Open-Source Post-Quantum Software

Douglas Stebila

UNIVERSITY OF WATERLOO

OPEN QUANTUM SAFE project

https://www.douglas.stebila.ca/research/presentations/

NSERC CRSNG

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

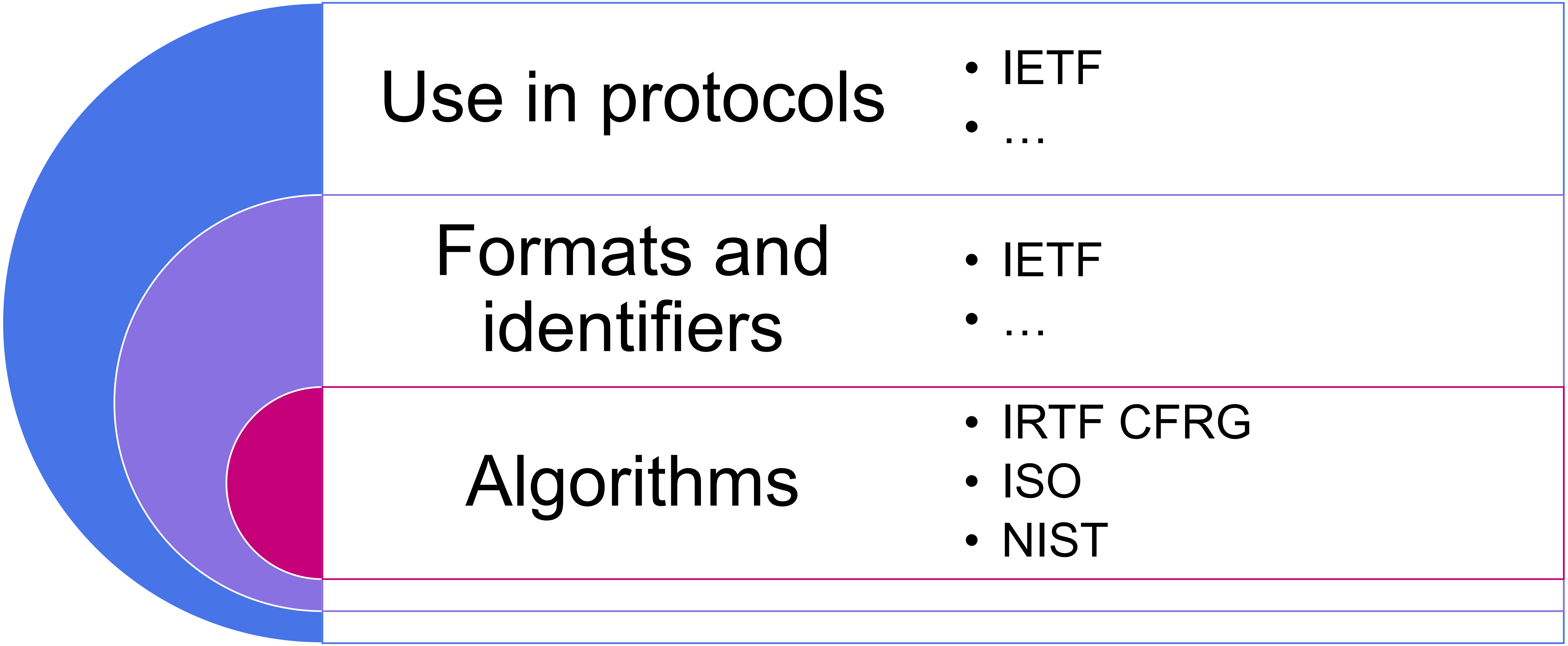
ICMC • 2023-09-21

Outline

1. Status of PQ standards
2. Open Quantum Safe project
3. New initiatives

Status of PQ Standards

Levels of standardization



Use in protocols

- IETF
- …

Formats and identifiers

- IETF
- …

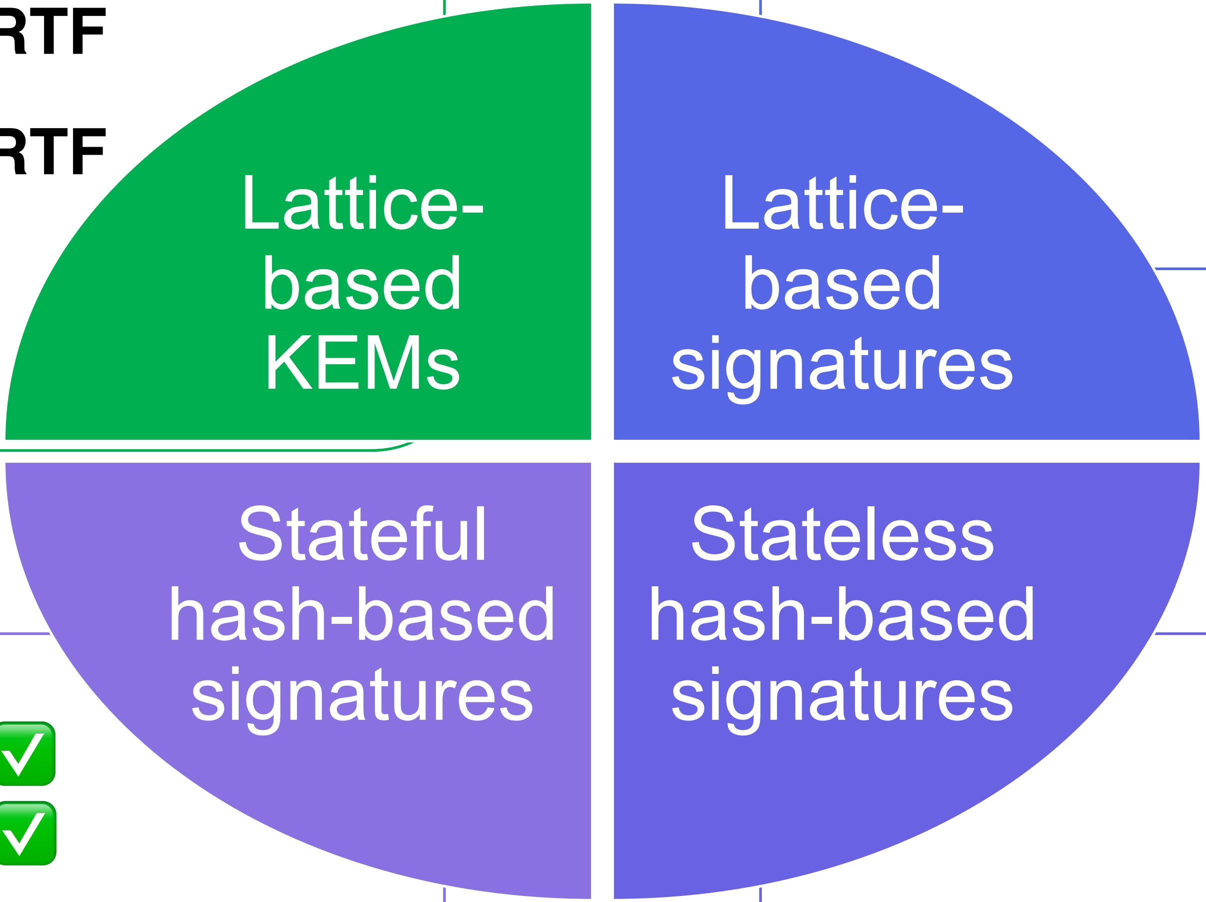
Algorithms

- IRTF CFRG
- ISO
- NIST

https://github.com/ietf-wg-pquip/state-of-protocols-and-pqc

PQ algorithms being standardized

Kyber	NIST	ISO	IRTF
NTRU			IRTF
FrodoKEM		ISO	
Classic McEliece		ISO	



Lattice-based KEMs

Lattice-based signatures

Stateful hash-based signatures

Stateless hash-based signatures

Dilithium	NIST
Falcon	NIST

XMSS
HSS/LMS
{ IRTF ✅
NIST ✅

SPHINCS+ NIST

What is “post-quantum TLS”?

Pre-shared key (PSK) mode

- Already implemented
- Still has key distribution problem
- No forward secrecy

What is “post-quantum TLS”?

Pre-shared key (PSK) mode	Key exchange	
	PQ-only	Hybrid
• Already implemented • Still has key distribution problem • No forward secrecy	• Fairly easy to implement • Needed soonest: harvest now, decrypt later	
		• Robust to 1 algorithm break • "Safe choice" • In demand during pre-certification

What is “post-quantum TLS”?

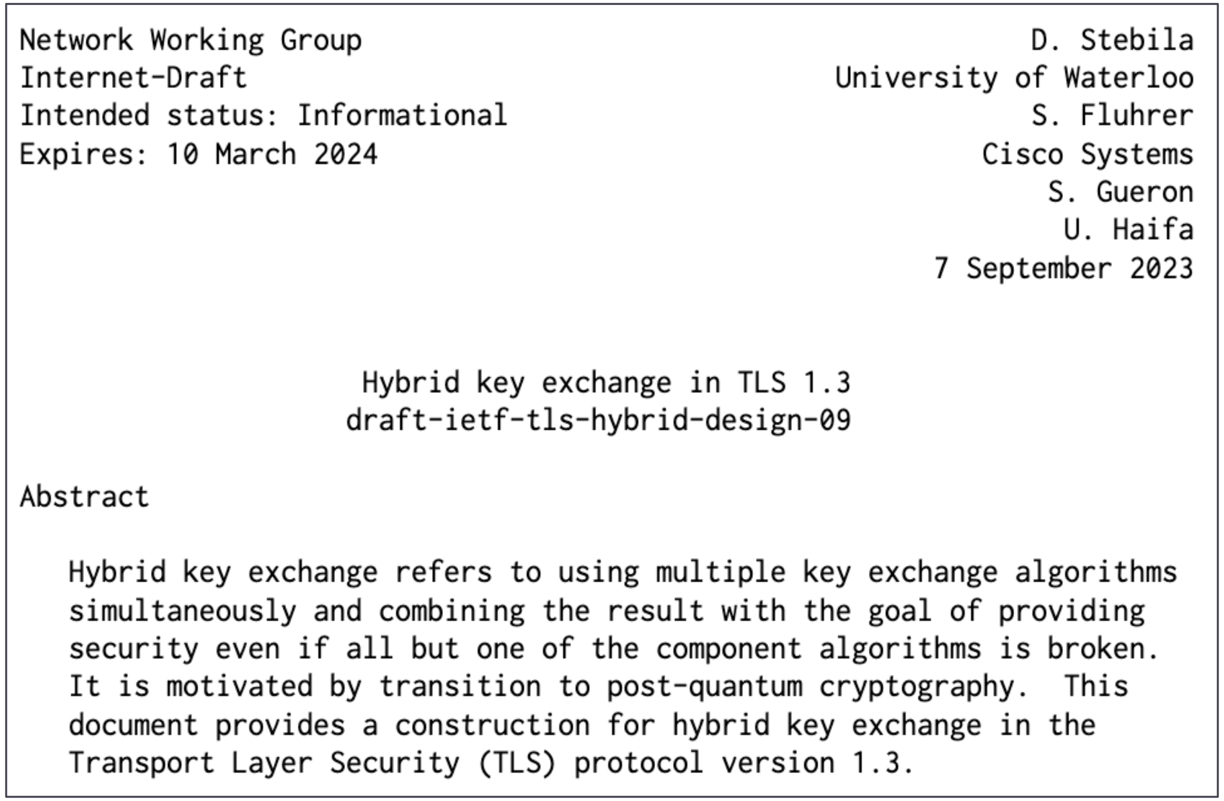
Pre-shared key (PSK) mode	Key exchange		Authentication	
	PQ-only	Hybrid	PQ-only	Hybrid / Composite
• Already implemented • Still has key distribution problem • No forward secrecy	• Fairly easy to implement • Needed soonest: harvest now, decrypt later		• Requires coordination with certificate authorities • Less urgently needed: can't retroactively break authentication • Size ☹	
		• Robust to 1 algorithm break • "Safe choice" • In demand during pre-certification		• May not make sense in the context of a negotiated protocol like TLS

What is “post-quantum TLS”?

Pre-shared key (PSK) mode	Key exchange		Authentication		Alternative protocol designs
	PQ-only	Hybrid	PQ-only	Hybrid / Composite	
• Already implemented • Still has key distribution problem • No forward secrecy	• Fairly easy to implement • Needed soonest: harvest now, decrypt later		• Requires coordination with certificate authorities Less urgently needed: can't retroactively break authentication • Size ☹		• e.g. AuthKEM / KEMTLS • Harder to implement; may require state machine changes • Lots of interesting research!
		• Robust to 1 algorithm break • "Safe choice" • In demand during pre-certification		• May not make sense in the context of a negotiated protocol like TLS	

Area of initial focus

Hybrid key exchange in TLS



```
Network Working Group                                        D. Stebila
Internet-Draft                                  University of Waterloo
Intended status: Informational                               S. Fluhrer
Expires: 10 March 2024                                    Cisco Systems
                                                              S. Gueron
                                                               U. Haifa
                                                       7 September 2023


                     Hybrid key exchange in TLS 1.3
                     draft-ietf-tls-hybrid-design-09

Abstract

   Hybrid key exchange refers to using multiple key exchange algorithms
   simultaneously and combining the result with the goal of providing
   security even if all but one of the component algorithms is broken.
   It is motivated by transition to post-quantum cryptography.  This
   document provides a construction for hybrid key exchange in the
   Transport Layer Security (TLS) protocol version 1.3.
```

- Fairly mature
- Early deployments showing reasonable performance:
  - Chrome
  - Cloudflare
  - Open Quantum Safe
  - WolfSSL
  - …
- Contains algorithm identifiers for Kyber768Draft00+x25519 and Kyber768Draft00+secp256r1

https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/

Algorithm standardization status

	Kyber	Dilithium	Falcon
Primary standardizer:	NIST	NIST	NIST
Status at NIST:	Draft available	Draft available	Draft standard pending
Status at IETF/IRTF:	CFRG draft available	No draft available	No draft available

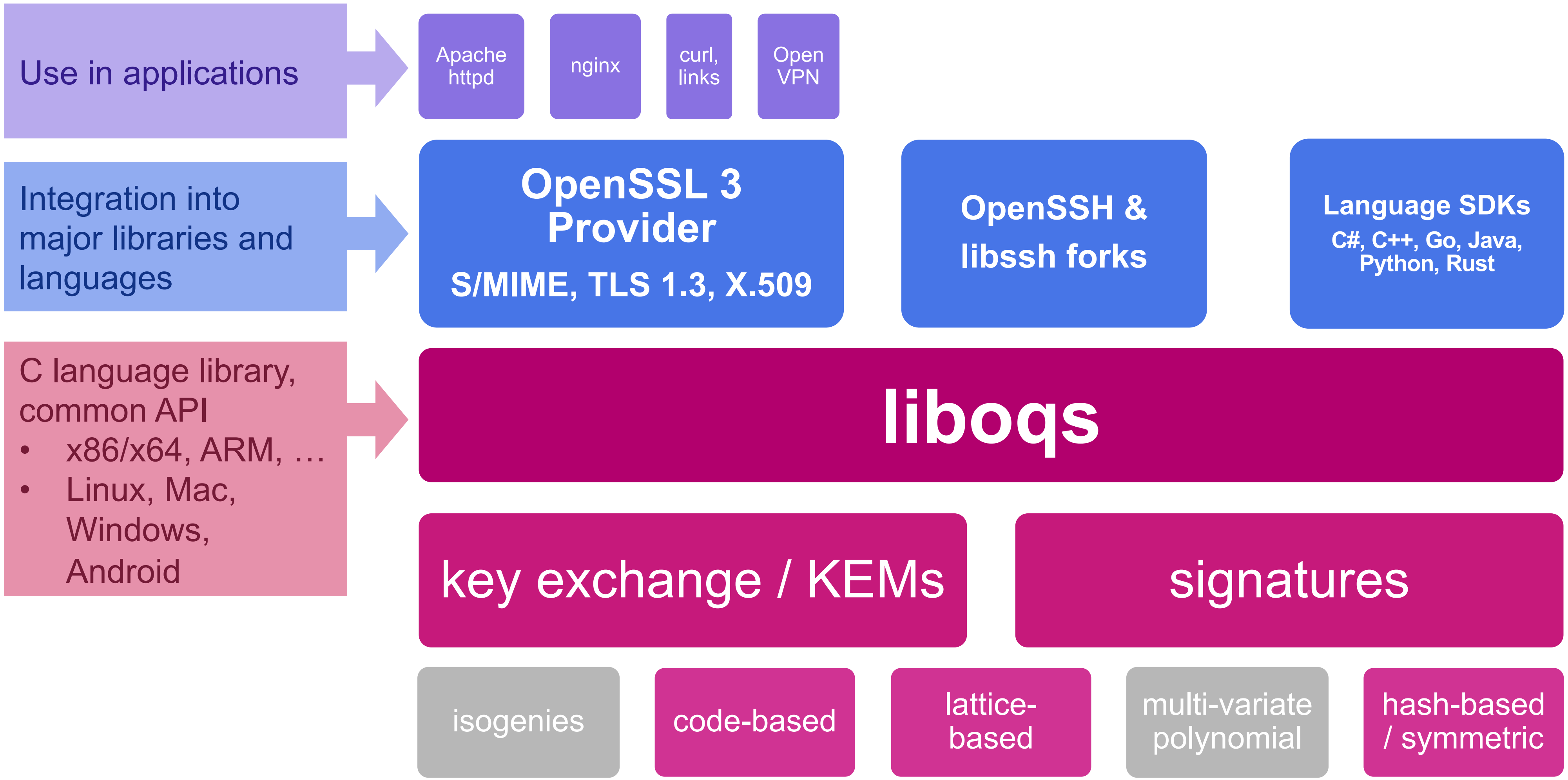
	SPHINCS+	XMSS	LMS
Primary standardizer:	NIST	IRTF	IRTF
Status at NIST:	Draft standard pending	Approved in SP 800-208 (2020)	Approved in SP 800-208 (2020)
Status at IETF/IRTF:	No draft available	RFC 8391 (2018)	RFC 8554 (2019) Draft for new parameter sets

Protocol	Key exchange / PKE	Authentication	Alternatives
TLS 1.3 (secure channel)	Drafts: Hybrid Kyber	Prototypes	• AuthKEM / KEMTLS • TurboTLS • Merkle Tree certs.
X.509 (certificates)	Drafts: • Identifiers for Kyber	Drafts: • Identifiers and formats for Dilithium, LMS, XMSS, SPHINCS+ • Composite keys and signatures • Threshold composite • Binding non-composite certs • IETF PQC PKI hackathon	
Secure Shell (SSH) (secure channel)	Drafts: Hybrid Kyber OpenSSH: Hybrid NTRU Prime	Prototypes	
IPsec (secure channel)	RFCs: PSK Drafts: hybrid, large messages	Drafts: • Hybrid non-composite • Negotiation	
CMS (secure email, …)	Drafts: KEMs, Kyber	RFCs: LMS Drafts: SPHINCS+	
DNSSEC (Domain Name Security)	Drafts: Stateful HBS		• Merkle Tree ladder • Request-based frag.
OpenPGP (secure email)	Drafts: • Composite Kyber	Drafts: • Composite Dilithium • PQ-only SPHINCS+	

https://github.com/ietf-wg-pquip/state-of-protocols-and-pqc

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

Open Quantum Safe Project



Use in applications

Apache httpd

nginx

curl, links

Open VPN

Integration into major libraries and languages

OpenSSL 3 Provider

S/MIME, TLS 1.3, X.509

OpenSSH & libssh forks

Language SDKs

C#, C++, Go, Java, Python, Rust

C language library, common API

- x86/x64, ARM, …
- Linux, Mac, Windows, Android

liboqs

key exchange / KEMs

signatures

isogenies

code-based

lattice-based

multi-variate polynomial

hash-based / symmetric

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

Led by University of Waterloo

Industry partners:

- Amazon Web Services
- Cisco
- evolutionQ
- IBM Research
- Microsoft Research
- SandboxAQ
- softwareQ

Additional contributors:

- Senetas
- PQClean project
- Individuals

Financial support:

- AWS
- Canadian Centre for Cyber Security
- Cisco
- NLNet
- NSERC
- Unitary Fund
- Verisign

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, …
- Includes NIST selections and all Round 4 candidates

https://openquantumsafe.org/liboqs/

liboqs current status

- Version 0.8.0 released June 2023
  - BIKE updated to Round 4
- Version 0.9.0 to be released in the next 2–3 weeks
  - Update Classic McEliece to Round 4
  - Build improvements on ARM, Windows
- Currently in branches for subsequent releases:
  - Updates to Kyber and Dilithium
    - Waiting on resolution of discrepancies between the PQ-Crystals Team's versions of Kyber & Dilithium and NIST's FIPS drafts
  - Development work on stateful hash-based signatures (XMSS and HSS/LMS)

https://openquantumsafe.org/liboqs/

Long term vision for liboqs

- Dual track: experimental + production
  - Experimental:
    - Continue to support testing of new algorithms in NIST Round 4 and Digital Signature On-Ramp
  - Production:
    - Move to formally verified or audited open source implementations of standardized algorithms

- Enlarge community of contributors
  - Working to build a permanent home for OQS

OpenSSL provider concept

Binary software crypto module API (shared lib: .so/.dll)

- Available since OpenSSL 3.0
- Allows addition of different/new implementations for encryption, signature, digesting, KEM, persistence (X.509), etc.
- Replacing OpenSSL 1 engine API

Core providers delivered with OpenSSL:

- default: Classic crypto (RSA, EC, AES, etc.)
- fips: Certified implementations of classic crypto
- legacy: Deprecated classic crypto

OpenSSL provider API

- OpenSSL core calls into providers to learn about & invoke their features

- Core/provider interface improving over time:
  - 3.0/3.1: Full TLS 1.3 KEM support; X.509 support
  - 3.2: Full TLS 1.3 signature support
  - 3.3+: Full PKCS#7 support

oqs-provider

- Uses liboqs to add all NIST-competition PQC KEM & SIG algorithms to OpenSSL 3+
- Further adds hybrid (PQ+classic) KEM & SIG

- Sample OpenSSL commands enabled:
  - genpkey/req/ca: X.509 cert generation, CA operation
  - s_server/s_client: TLS1.3 KEM & signature server
  - cms, dgst, verify
- Use in OpenSSL-reliant applications:
  - Curl, httpd, nginx, openvpn, epiphany, ...

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

Getting oqs-provider

- Source:
  - https://github.com/open-quantum-safe/oqs-provider
  - Latest release: 0.5.1
- Docker image:
  - Ubuntu-based, OpenSSL 3.2, oqsprovider-enabled
  - Also available for interop testing
- Binaries (shared libs) for x64:
  - .deb (Debian)
  - .dylib (MacOS)
  - .dll (Windows)

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

Deployment of oqs-provider

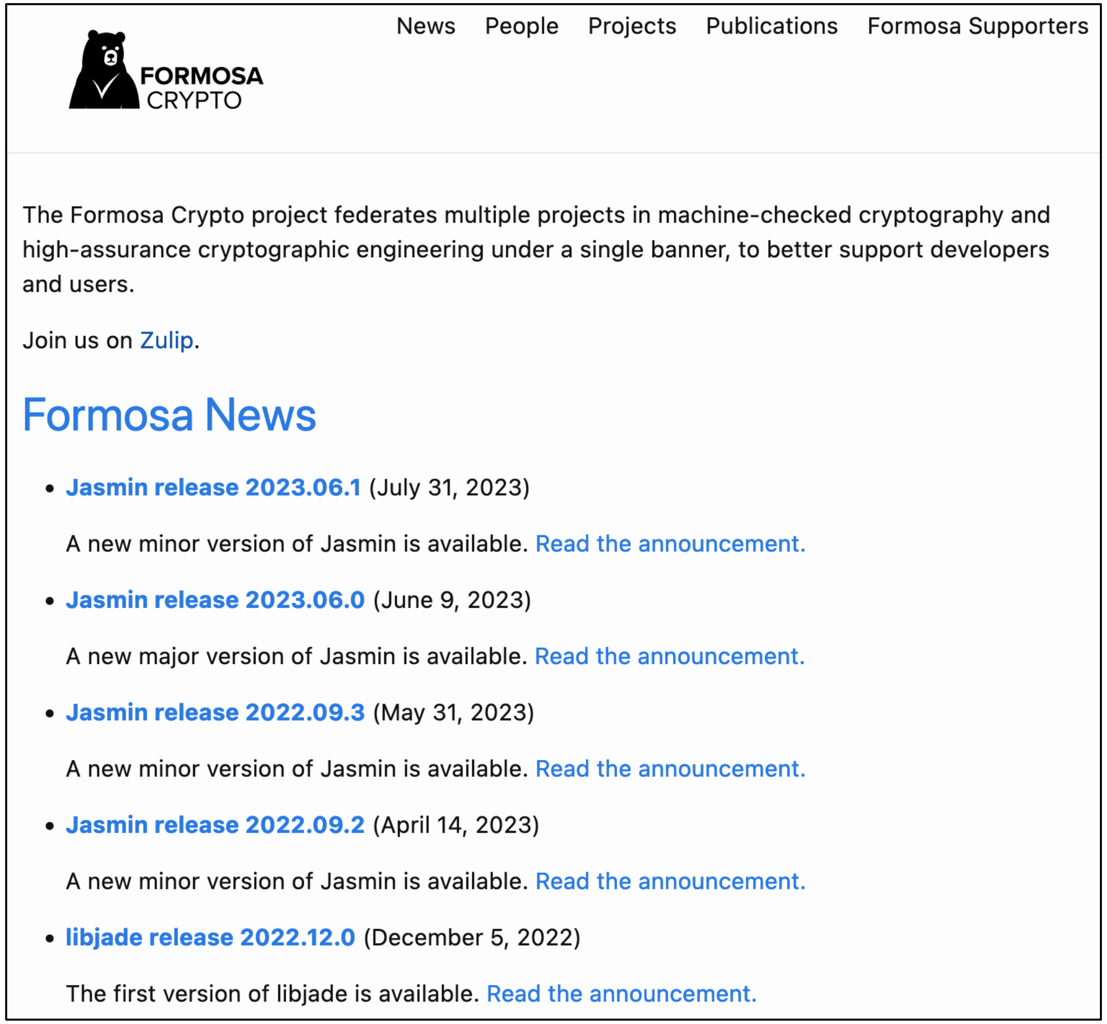
- Interop test server for
  - X.509 PQ & hybrid certificates
  - PQ & hybrid TLS1.3 operations
  - https://test.openquantumsafe.org

- Matrix of single ports permitting use of
  - [RSA|EC +] {Dilithium, Falcon, SPHINCS+ } &&
  - [RSA|EC +] {Kyber, Frodo, HQC, Bike }

https://test.openquantumsafe.org

New initiatives

Formosa Crypto*



The Formosa Crypto project federates multiple projects in machine-checked cryptography and high-assurance cryptographic engineering under a single banner, to better support developers and users.

Join us on Zulip.

Formosa News

- **Jasmin release 2023.06.1** (July 31, 2023)

  A new minor version of Jasmin is available. Read the announcement.
- **Jasmin release 2023.06.0** (June 9, 2023)

  A new major version of Jasmin is available. Read the announcement.
- **Jasmin release 2022.09.3** (May 31, 2023)

  A new minor version of Jasmin is available. Read the announcement.
- **Jasmin release 2022.09.2** (April 14, 2023)

  A new minor version of Jasmin is available. Read the announcement.
- **libjade release 2022.12.0** (December 5, 2022)

  The first version of libjade is available. Read the announcement.

https://formosa-crypto.org/

Focuses on machine-checked cryptographic proofs and implementations

- **EasyCrypt**: Tool for verification of game-based cryptographic proofs
- **Jasmin**: Language for high-assurance cryptographic implemetnations
- **libjade**: Cryptographic library in Jasmin with proofs in EasyCrypt

* I'm not involved in Formosa Crypto, just a fan ☺

EasyCrypt / Jasmin / libjade toolchain

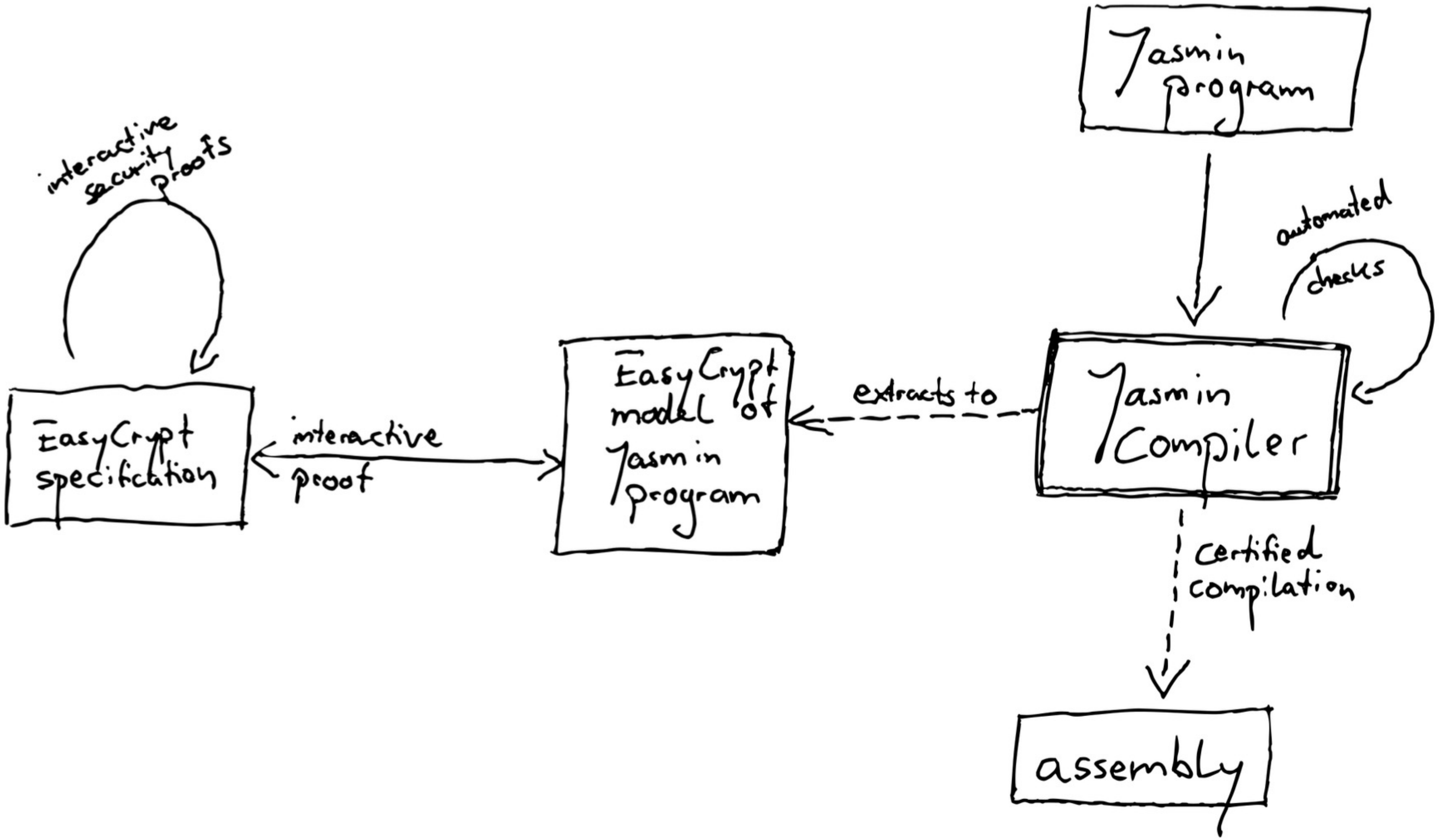


Diagram by Peter Schwabe https://cryptojedi.org/peter/data/ches-20230911.pdf

Starting later 2023: Kyber code package

- **Goal**: high-assurance* open-source implementations of Kyber for a variety of target architectures and languages distributed primarily as source code for other cryptographic libraries and tools to incorporate

  *High-assurance: formally verified, audited, or certified

- Looking for community involvement! Contact me or Peter Schwabe

New Initiatives in Open Source Post-Quantum Software

Douglas Stebila

UNIVERSITY OF WATERLOO

OPEN QUANTUM SAFE

https://www.douglas.stebila.ca/research/

Open Quantum Safe project

- liboqs
  - KEMs: Kyber, Round 4 candidates
  - SIGs: Dilithium, Falcon, SPHINCS+
- oqs-provider for OpenSSL 3
  - PQ + hybrid certificates
  - PQ + hybrid TLS 1.3 key exchange and signatures

Get involved!

- OQS community growing
- Kyber code package coming in 2024

https://openquantumsafe.org/