CRYSTALS–Kyber

Roberto Avanzi, Joppe Bos, Jintai Ding, Léo Ducas, Eike Kiltz, Tancrède Lepoint,
Vadim Lyubashevsky, John M. Schanck, **Peter Schwabe**, Gregor Seiler, Damien Stehlé

authors@pq-crystals.org

https://pq-crystals.org/kyber

November 29, 2022

Kyber summary

- MLWE-based IND-CCA2-secure KEM
  - IND-CPA secure LPR public-key encryption
  - Tweaked FO transform
- Only KEM selected by NIST for standardization after round 3

Kyber summary

- MLWE-based IND-CCA2-secure KEM
  - IND-CPA secure LPR public-key encryption
  - Tweaked FO transform
- Only KEM selected by NIST for standardization after round 3
- Very fast across different platforms
- E.g., $\approx 2\times$ faster than X25519 on Skylake (at level 3)
- Will be even faster with HW Keccak acceleration

Kyber summary

- MLWE-based IND-CCA2-secure KEM
  - IND-CPA secure LPR public-key encryption
  - Tweaked FO transform
- Only KEM selected by NIST for standardization after round 3
- Very fast across different platforms
- E.g., $\approx 2\times$ faster than X25519 on Skylake (at level 3)
- Will be even faster with HW Keccak acceleration
- Same optimized routines across all parameter sets
- Designed for efficient constant-time implementation
- Designed for efficient vectorization
- Designed for low memory consumption on embedded platforms

Decisions I: symmetric crypto

Current symmetric crypto

H:	SHA3-256
G:	SHA3-512
PRF:	SHAKE-256
KDF:	SHAKE-256
XOF:	SHAKE-128

Possible alternative

H:	cSHAKE-256
G:	cSHAKE-256
PRF:	cSHAKE-256
KDF:	cSHAKE-256
XOF:	SHAKE-128

Decisions I: symmetric crypto

Current symmetric crypto

H:	SHA3-256
G:	SHA3-512
PRF:	SHAKE-256
KDF:	SHAKE-256
XOF:	SHAKE-128

Possible alternative

H:	cSHAKE-256
G:	cSHAKE-256
PRF:	cSHAKE-256
KDF:	cSHAKE-256
XOF:	SHAKE-128 . . . or

TurboKeccak

- XOF is used to generate public matrix A
- 12-round Keccak sufficient as secure hash function
- Don't even need full-fledged hash function for generating A

Decisions II: FO transform

Hashing prefix(pk)

- Kyber hashes H(pk) into coins and shared key
  - Protection against multitarget failure attacks
  - Makes KEM "contributory"
- Cheaper and sufficient: Use prefix(pk) instead

Decisions II: FO transform

Hashing prefix(pk)

- Kyber hashes H(pk) into coins and shared key
  - Protection against multitarget failure attacks
  - Makes KEM "contributory"
- Cheaper and sufficient: Use prefix(pk) instead

Ciphertext hash

- Kyber hashes H(c) into shared key
- "Robust": shared key depends on full transcript

Decisions II: FO transform

Hashing prefix(pk)

- Kyber hashes H(pk) into coins and shared key
  - Protection against multitarget failure attacks
  - Makes KEM "contributory"
- Cheaper and sufficient: Use prefix(pk) instead

Ciphertext hash

- Kyber hashes H(c) into shared key
- "Robust": shared key depends on full transcript
- Not useful in proofs of any security property
- Complicates QROM proofs
- Dropping this hash would simplify QROM proofs and speed up Encaps

Deployment examples

- All websites and APIs served by Cloudflare; see https://blog.cloudflare.com/post-quantum-for-all/
- TLS 1.3 With X25519+Kyber512 in Firefox by Tamvada; see https://github.com/xvzcf/firefox-pq-demos

Deployment examples

- All websites and APIs served by Cloudflare; see
  https://blog.cloudflare.com/post-quantum-for-all/
- TLS 1.3 With X25519+Kyber512 in Firefox by Tamvada; see
  https://github.com/xvzcf/firefox-pq-demos
- AWS Secrets Manager using TLS with Kyber; see
  https://aws.amazon.com/about-aws/whats-new/2022/08/
  aws-secrets-manager-connections-support-hybrid-post-quantum-tls-kyber/

Deployment examples

- All websites and APIs served by Cloudflare; see https://blog.cloudflare.com/post-quantum-for-all/
- TLS 1.3 With X25519+Kyber512 in Firefox by Tamvada; see https://github.com/xvzcf/firefox-pq-demos
- AWS Secrets Manager using TLS with Kyber; see https://aws.amazon.com/about-aws/whats-new/2022/08/aws-secrets-manager-connections-support-hybrid-post-quantum-tls-kyber/
- IBM quantum-secure tape drive; see https://www.ibm.com/blogs/research/2019/08/crystals/
- IBM Cloud key management; see https://www.ibm.com/cloud/blog/introducing-quantum-safe-crypto-tls-for-ibm-key-protect

Implementations

- Kyber GitHub repo (C ref and AVX2): https://github.com/pq-crystals/kyber

Implementations

- Kyber GitHub repo (C ref and AVX2): https://github.com/pq-crystals/kyber
- PQClean (C ref and AVX2): https://github.com/PQClean/PQClean

Implementations

- Kyber GitHub repo (C ref and AVX2): https://github.com/pq-crystals/kyber
- PQClean (C ref and AVX2): https://github.com/PQClean/PQClean
- pqm4 (C/asm for Arm Cortex-M4): https://github.com/mupq/pqm4

Implementations

- Kyber GitHub repo (C ref and AVX2): https://github.com/pq-crystals/kyber
- PQClean (C ref and AVX2): https://github.com/PQClean/PQClean
- pqm4 (C/asm for Arm Cortex-M4): https://github.com/mupq/pqm4
- libjade (jasmin $\rightarrow$ asm): https://github.com/formosa-crypto/libjade

Implementations

- Kyber GitHub repo (C ref and AVX2): https://github.com/pq-crystals/kyber
- PQClean (C ref and AVX2): https://github.com/PQClean/PQClean
- pqm4 (C/asm for Arm Cortex-M4): https://github.com/mupq/pqm4
- libjade (jasmin → asm): https://github.com/formosa-crypto/libjade
- Incomplete list of third-party implementations:
  https://pq-crystals.org/kyber/software.shtml

SCA/FI attacks and countermeasures

- Baseline: all(?) implementations are constant time
- Protections against Spectre v1 [ABGLOPST22]: https://ia.cr/2022/1270

SCA/FI attacks and countermeasures

- Baseline: all(?) implementations are constant time
- Protections against Spectre v1 [ABGLOPST22]: https://ia.cr/2022/1270
- Numerous papers on HW SCA and FI, see, e.g.,
  - survey + new results [RCDB22]: https://ia.cr/2022/737
  - attacks against higher-order masked *Saber* [NWDP22]: https://ia.cr/2022/919

SCA/FI attacks and countermeasures

- Baseline: all(?) implementations are constant time
- Protections against Spectre v1 [ABGLOPST22]: https://ia.cr/2022/1270
- Numerous papers on HW SCA and FI, see, e.g.,
  - survey + new results [RCDB22]: https://ia.cr/2022/737
  - attacks against higher-order masked *Saber* [NWDP22]: https://ia.cr/2022/919
- Also numerous papers on countermeasures, see, e.g.,
  - first and higher-order masking by [BGRSvV21]: https://ia.cr/2021/483
  - combined SCA and FI countermeasures by [HP21]: https://ia.cr/2021/101

SCA/FI attacks and countermeasures

- Baseline: all(?) implementations are constant time
- Protections against Spectre v1 [ABGLOPST22]: https://ia.cr/2022/1270
- Numerous papers on HW SCA and FI, see, e.g.,
  - survey + new results [RCDB22]: https://ia.cr/2022/737
  - attacks against higher-order masked *Saber* [NWDP22]: https://ia.cr/2022/919
- Also numerous papers on countermeasures, see, e.g.,
  - first and higher-order masking by [BGRSvV21]: https://ia.cr/2021/483
  - combined SCA and FI countermeasures by [HP21]: https://ia.cr/2021/101
- No consensus/understanding on "sufficient" countermeasures; see, e.g. https://iacr.org/submit/files/slides/2022/rwc/rwc2022/48/slides.pdf
- Much more work required – need for coordination?

Kyber online

https://pq-crystals.org/kyber