CRYSTALS–Kyber

Roberto Avanzi, Joppe Bos, 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

June 9, 2021

Reminder: the big picture

Kyber.CPAPKE: LPR encryption or "Noisy ElGamal"

$\mathbf{s}, \mathbf{e} \leftarrow \chi$

$sk = \mathbf{s}, pk = \mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$

$\mathbf{r} \leftarrow \chi$

$\mathbf{e}_1, e_2 \leftarrow \chi'$

$\mathbf{u} \leftarrow \mathbf{A}^T\mathbf{r} + \mathbf{e}_1$

$v \leftarrow \mathbf{t}^T\mathbf{r} + e_2 + \mathsf{Enc}(m)$

$c = (\mathbf{u}, v)$

$m = Dec(v - \mathbf{s}^T\mathbf{u})$

Reminder: the big picture

Kyber.CCAKEM: CCA-secure KEM via tweaked FO transform

- Use implicit rejection
- Hash public key into seed and shared key
- Hash ciphertext into shared key
- Use Keccak-based functions for all hashes and XOF

Reminder: the big picture

Kyber.CCAKEM: CCA-secure KEM via tweaked FO transform

- Use implicit rejection
- Hash hash of public key into seed and shared key
- Hash hash of ciphertext into shared key
- Use Keccak-based functions for all hashes and XOF

Changes and updates since round 2

Changes affecting testvectors

- Increase noise for level-1 parameter set
- Reduce ciphertext compression for level-1 parameter set
- More efficient uniform sampling of **A**

Changes and updates since round 2

Changes affecting testvectors

- Increase noise for level-1 parameter set
- Reduce ciphertext compression for level-1 parameter set
- More efficient uniform sampling of **A**

Other changes

- More detailed concrete security analysis
- Updated performance numbers

Security of Kyber512

- Discussion about Kyber512 classical gate-count security
- Started by Bernstein (20200530001531.21905.qmail@cr.yp.to)

Security of Kyber512

- Discussion about Kyber512 classical gate-count security
- Started by Bernstein (20200530001531.21905.qmail@cr.yp.to)
- Two questions raised/discussed:
  - Do classical attacks against Kyber512 require $\leq 2^{143}$ gates?
  - How relevant is gate count metric (*"debunked metric"*)?

Updates to Kyber512

- Wider distribution for $\mathbf{s}, \mathbf{e}$, and $\mathbf{r}$
- In Encaps additional "LWR" noise from compression
- Reduce ciphertext compression to control failure prob.

Updates to Kyber512

- Wider distribution for $\mathbf{s}, \mathbf{e}$, and $\mathbf{r}$
- In Encaps additional "LWR" noise from compression
- Reduce ciphertext compression to control failure prob.
- Analyze concrete security for (LWE+LWR)
- core-SVP hardness:
  - 112 bits under LWE assumption (same as round-2)
  - 118 bits under LWE+LWR assumption

Beyond Core-SVP hardness

- Gate count analysis for attacks against Kyber512:
  - Focus on primal attack
  - Use progressive BKZ
  - Take into account dimensions-for-free (D4F) optimization
  - Current understanding of gate cost of sieving
- Tentative gate count of $2^{151.5}$

Beyond Core-SVP hardness

- Gate count analysis for attacks against Kyber512:
  - Focus on primal attack
  - Use progressive BKZ
  - Take into account dimensions-for-free (D4F) optimization
  - Current understanding of gate cost of sieving
- Tentative gate count of $2^{151.5}$
- Detailed discussion of approximations, overheads and foreseeable improvements
- Conclusion: gate count in $[2^{135.5}, 2^{165.5}]$
- Details: See Section 5.2 of the Kyber specification

Observations about FO tweaks

- Hashing H(pk) into coins: multitarget protection
  - Cheaper approach: talk in ≈2h by Julien Duman

Observations about FO tweaks

- Hashing $\mathsf{H}(pk)$ into coins: multitarget protection
  - Cheaper approach: talk in $\approx$2h by Julien Duman
- Hashing $\mathsf{H}(c)$ into final key ($K := \mathsf{KDF}(\bar{K} \| \mathsf{H}(c))$)
  - Shared key depends on full transcript
  - Use $\mathsf{H}(c)$ for non-incremental hash APIs

Observations about FO tweaks

- Hashing H(pk) into coins: multitarget protection
  - Cheaper approach: talk in ≈2h by Julien Duman
- Hashing H(c) into final key ($K := \mathsf{KDF}(\bar{K} \| \mathsf{H}(c))$)
  - Shared key depends on full transcript
  - Use H(c) for non-incremental hash APIs
- Earlier talk by Varun Maram:
  - $K := \mathsf{KDF}(\bar{K} \| \mathsf{H}(c))$ tricky to prove in the QROM
  - $K := \mathsf{KDF}(\bar{K} \| c)$ would be fine

Observations about FO tweaks

- Hashing H(pk) into coins: multitarget protection
  - Cheaper approach: talk in ≈2h by Julien Duman
- Hashing H(c) into final key ($K := \mathsf{KDF}(\bar{K} \| \mathsf{H}(c))$)
  - Shared key depends on full transcript
  - Use H(c) for non-incremental hash APIs
- Earlier talk by Varun Maram:
  - $K := \mathsf{KDF}(\bar{K} \| \mathsf{H}(c))$ tricky to prove in the QROM
  - $K := \mathsf{KDF}(\bar{K} \| c)$ would be fine
- Lots of new results on FO in the last 4 years
- Revisit details of FO during standardization (inpendent of chosen scheme(s))?

Other ongoing work (selection)

- Formal verification of Kyber (and Saber)
  - Earlier talk by Matthias Meijers

Other ongoing work (selection)

- Formal verification of Kyber (and Saber)
  - Earlier talk by Matthias Meijers
- Side-channel protection (masking) of Kyber
  - Earlier talk by Daniel Heinz
  - Earlier talk by Michiel van Beirendonck
  - Bos, Gourjon, Renes, Schneider, van Vredendaal https://eprint.iacr.org/2021/483

Other ongoing work (selection)

- Formal verification of Kyber (and Saber)
  - Earlier talk by Matthias Meijers
- Side-channel protection (masking) of Kyber
  - Earlier talk by Daniel Heinz
  - Earlier talk by Michiel van Beirendonck
  - Bos, Gourjon, Renes, Schneider, van Vredendaal https://eprint.iacr.org/2021/483
- Optimized implementations
  - Earlier talk by Kris Gaj (FPGA)
  - Earlier talk by Duc Tri Nguyen

Kyber online

https://pq-crystals.org/kyber