Overview of results on KpqC Round 2 candidates

Daniel J. Bernstein, Jolijn Cottaar,
Emanuele Di Giandomenico, Kathrin Hövelmanns,
Andreas Hülsing, Mikhail Kudinov, Tanja Lange,
Mairon Mahzoun, Matthias Meijers. Alex Pellegrini,
Alberto Ravagnani, Silvia Ritsch, Sven Schäge,
Tianxin Tang, Monika Trimoska,
Marc Vorstermans, and Fiona Johanna Weber

Eindhoven University of Technology & University of Illinois at Chicago
authors ordered alphabetically

October 22, 2024

Our approach for Round 2

- Submissions in 2nd round much more mature
  (by selection of solid candidates and time passing).
- We proceeded with cryptanalysis, analysis of proofs, and analysis of implementations in parallel.
  (Also thanks to a much bigger team).
- Emails on KpqC mailing list about implementation and benchmarking issues. Many submissions got updated and re-benchmarked.
  Some changes are still coming in.
- Emails about issues in proofs only to teams, trying to offer assistance.

KEMs

KEMs – NTRU+

- Lattice-based KEM, related to NTTRU, uses cyclotomic polynomials of degree $2^i 3^j$ for better scaling and NTT.
- Avoids re-encryption by recovering randomness and having r =hash(m^+) and m =SOTP(m^+,hash(r)). ,
- New version has update in SOTP transform (deals with IND-CCA2 attack from Round 1), but uses new definition of rigidity.
- Round-2 software:
  - Easy to integrate into SUPERCOP.
  - Correctness issue identified. Fixed by NTRU+ team.
  - No timing leaks identified.

KEMs – PALOMA

- Code-based KEM using Goppa codes.
- Avoids re-encryption.
- Reaction attack using software error.
- Reaction attack using generalized Patterson decoder.
- Both would have been caught by re-encryption.
- Partial key recovery due to special handling of 0 in support by generalized Patterson.
- September: updated submission with new proof & handling of weight $< t$ seems to fix issues.
- Round-2 software issues:
  - Permutations use Fisher–Yates sampling.
  - Decoding algorithm uses variable-degree polynomials.
  - Software uses lookup tables for field arithmetic.
  - PALOMA team sent new software 2 days ago.

KEMs – REDOG

- Code-based KEM using rank-metric codes.
- Security proof having some gaps.
- Message recovery attack uses $t_2 \in \{2, 3\}$ to search through all arrangements of the 2^{t_2} elements over all positions at cost $2^{t_2(n-k)}$.
  For each arrangement a linear-algebra attack is tried that succeeds if the arrangement is correct.

Security	$(n, k, \ell, q, m, r, \lambda, t_1, t_2)$	Pad Thai attack
128	$(30, 6, 25, 2, 59, 12, 3, 6, 2)$	**87.92**
192	$(44, 8, 37, 2, 83, 18, 3, 12, 2)$	**132**
256	$(58, 10, 49, 2, 109, 24, 3, 15, 3)$	**230.53**

- Round-2 software:
  - Still no functioning C implementation.
  - We did a Python implementation.

KEMs – SMAUG-T

- Lattice-based KEM using module lattices, smaller than Kyber.
- System results from merger SMAUG and TiGER, most parameter sets match SMAUG and TiMER variant keeps the small ciphertext size of TiGER.
- SMAUG-T continues to avoid error correction for main parameter choices.
- Complete proof (just minor comments).
- Round-2 software:
  - Exploitable correctness issues.
  - Exploitable timing leaks: e.g., Fisher–Yates sampling.
  - Several recent software updates.

Signatures

Signatures – AIMer

- Signature based on MPCitH.
- Uses symmetric primitive AIM as one-way function.
- Round 1 saw several attacks on AIM using algebraic attacks.
- Proof follows known framework and has fixed some issues.
- Issues are lack of some explanations for parameters and intuition for the proof. Lots of algebraic structure that can possibly still be exploited.
- Round-2 software:
  - Some correctness issues.
  - Some timing variations that are presumably exploitable: table lookups for field multiplication. (In reference software; AVX2 software avoids this.)

Signatures – HAETAE

- Lattice-based signature using module lattices.
- Close to Dilithium but uses different error distribution (hyberball sampling).
- Kim and later Lee, Ryu, and Lee found issue with parameter choices (now fixed).
- Solid security proof (we had very few comments), issues from Round 1 have been addressed.
- Round-2 software:
  - Most timing variations look like rejection sampling.
  - One division is presumably exploitable, now fixed.

Signatures – MQ-Sign

- Signature following UOV design (multivariate system).
- Serious security issues in 3 out of 4 cases for round 1.
- Round 2 keeps unbroken one; introduced new LR variant (dropping sparseness but keeping circulant structure).
- For LR-variant can sign 0 without private key.
- Proof refers to other paper without showing details of how this matches, system does not use full-domain hash
- Round-2 software:
  - Some correctness issues, maybe exploitable.
  - Some timing leaks, maybe exploitable.

Signatures – NCC-Sign

- Lattice-based signature using ideal lattices
- Related to NTRU Prime and NTTRU; implemented version uses NTTRU's cyclotomic trinomials.
- Proof refers to Dilithium and had same issue in round 1.
- Round-2 proof still mostly refers to other papers and has some gaps that should be fixable.
- Round-2 software:
  - Seems to cover only the trinomial case.
  - Some correctness issues, presumably exploitable.
  - Many timing leaks: e.g., Fisher–Yates sampling.

Benchmarking