FIPS 203: **Module-Lattice-based Key-Encapsulation Mechanism (ML-KEM)**

Brief timeline:
- **07/22**: NIST PQC Round 3 ends
- **12/22**: Kyber team proposes changes to Kyber v3.02, asks community for feedback
  - Choice of symmetric primitives in Kyber
  - Modify FO transform
- **04/23**: NIST incorporates FO modification into draft FIPS 203, asks community for feedback
- **08/23**: Draft FIPS 203 released, request for public comment
Public comment period

- 42 commenters (submitting 90 pages)
  - Some comments common to FIPS 203, 204, 205
  - Some comments unique to FIPS 203
  - All comments are available on the NIST PQC Project page

- Lots of forum posts
Auxiliary functions
- Choice of hash functions/XOFs (e.g., SHA2 vs SHA3)

Spec/guidance
- SHAKE API (for using it as a byte stream)
- Implementation guidance (bytes vs bits, test vectors, etc.)
- Testing/validation
- Security strength categories

Editorial
- Unify language and notation across FIPS 203 and 204
- Clarify various pieces of text
Core algorithms
1. Revert fully to Kyber v3.0 (combines changes 2-5)
2. Revert FO change (reintroduce hash of ciphertext)
3. KeyGen: revert indexing of A-matrix
4. Encaps: reintroduce hash of RNG output
5. Encaps: don't validate public key
6. Decaps: switch to explicit rejection
7. Decaps: change order of inputs to J() in Step 7 (see later slides)

Spec/guidance
- Allow storing keys as seeds
- Update 56C to support use of ML-KEM
- Provide more guidance on KEMs and their usage

Parameter sets
- Remove Kyber-512 entirely
NIST\textbf{ does} plan to do the following in FIPS 203:

1. **Revert A-matrix indexing** (minor but compatibility-breaking change)

2. **Specify XOF API** (for SHAKE)
   - Three operations: \textit{Initialize}, \textit{Absorb}, \textit{Squeeze}
   - Rewrite \textit{SampleNTT} to use this API
   \textbf{Why?} Existing standards did not allow using SHAKE as a stream

3. **Specify "lower-level" derandomized API**
   - "top-level" API remains the same (i.e., randomized \textit{KeyGen} and \textit{Encaps})
   - each top-level algorithm validates inputs, then runs RNG, then calls low-level algorithm
   \textbf{Why?} Enables CAVP testing: KATs well-defined and allows storing keys as seeds

\textbf{Feedback requested on this!}
Recall Step 7 in *Decaps*: $J(z || c)$.

- **Comment 1**: change to $J(c || z)$ so that masking the permutations on $c$ is not needed (a)
- **Comment 2**: change to $J(z || H(c))$ as an alternative (b)

- Revealing $z$ makes *Decaps* become explicit rejection.
- (b) computes 1 permutation more than (a) for ML-KEM-768.
- Both options require masking only 1 permutation.

We welcome your comments/input.
NIST plans to do the following to support FIPS 203:

1. Current Key Validation
   - Encaps and Decaps presently do input validation
   - Additional text and guidance will be in SP 800-227
   - We are still discussing internally

2. KDFs and KEM Combiners
   - KDFs of SP 800-56C can be applied to shared secrets (K) generated as specified in FIPS 203
   - More guidance for (IND-CCA2) hybrid KEMs will be provided in the forthcoming SP 800-227
   - We are still discussing internally
Planned rejections

NIST does **NOT** plan to do any of the following:

1. Give general KEM guidance in FIPS 203 (see forthcoming SP 800-227 instead)
2. Remove ML-KEM-512
3. Reintroduce hash of RNG output in *Encaps*
4. Revert FO change (reintroduce hash of ciphertext)
5. Switch to explicit rejection in *Decaps*
6. Revert to Kyber v3.0

(See forum for discussions of pros/cons.)
Thank you

Please share your comments and suggestions!

Send comments to pqc-comments@nist.gov
Public discussions: pqc-forum@list.nist.gov