Post-Quantum Cryptography in OpenPGP with NIST and Brainpool EC Domain Parameters

draft-ehlen-openpgp-nist-bp-comp

Quynh Dang [NIST] Stephan Ehlen[BSI], Johannes Roth[MTG], Falko Strenzke[MTG],

BSI: Federal Office for Information Security, Germany
MTG: MTG AG, Germany
NIST: National Institute of Standards and Technology, USA

draft-ehlen-openpgp-nist-bp-comp

- https://datatracker.ietf.org/doc/draft-ehlen-openpgp-nist-bp-comp
- public repository
  - https://github.com/openpgp-pqc/draft-ehlen-openpgp-nist-bp-comp

New code points for NIST and Brainpool EC Domain Parameters with ML-*

all code points are “MAY”

NIST	**Brainpool**
ML-KEM-512+NIST-P-256	—
ML-KEM-768+NIST-P-384	ML-KEM-768+brainpoolP256r1
ML-KEM-1024+NIST-P-384	ML-KEM-1024+brainpoolP384r1
ML-DSA-44+NIST-P-256	—
ML-DSA-65+NIST-P-384	ML-DSA-65+brainpoolP256r1
ML-DSA-87+NIST-P-384	ML-DSA-87+brainpoolP384r1

Rationales:	
Pair equivalent levels. NIST-P-384 is CNSA approved for all classification levels.[1] 521 bit curve not widely used.	Security margin for module-lattice schemes. 512 bit curve not widely used.

[1] CNSA 2.0 specification by NSA.

Further remarks

- PQC protocol aspects are implictly adopted from draft-ietf-openpgpg-pqc
- KEM combiner is equal to draft-ietf-openpgpg-pqc, but repeated verbatim

Questions

- Approval of proposed code points?
- Should we have “MUST” code points?
- Somewhat editorial: Should the KEM combiner only be referenced? (and not be repeated verbatim)

Upcoming changes

- KEM combiner: NIST compliance
  - upcoming change: KDF input order will change as in draft-ietf-openpgpg-pqc
  - still in discussion[2]: compliance problem due to KDF based on SHA3 only being qualified to receive the raw ECDH coordinate – no prior hashing allowed as in the current construction
- final version of draft-ehlen and draft-ietf-openpgpg-pqc will use ML-DSA and SLH-DSA pre-hash variants
  - crypto-refresh errs when using PureEdDSA, should have used HashEdDSA (no security implication, though)
  - error seems to be founded on misinterpretation of $H(m)$ in the algorithm description: $H(m)$ is the input to the signature function. Reinterpretation of the hash as the message is not allowed.

[2] https://github.com/openpgp-pqc/draft-openpgp-pqc/issues/132, https://github.com/openpgp-pqc/draft-ehlen-openpgp-nist-bp-comp/issues/10#issuecomment-2220090284