Classic McEliece: conservative code-based cryptography: modifications for round 2

30 March 2019

NIST has requested more security levels, and we would like to illustrate the flexibility of Classic McEliece. We are therefore expanding the list of parameter sets (n, k, t) as follows:

- $(8192, 6528, 128)$, 240-byte ciphertexts: taking both n and t to be powers of 2; as in round-1 submission.
- $(6960, 5413, 119)$, 226-byte ciphertexts: optimal security within 2^{20} bytes for public key; as in round-1 submission.
- $(6688, 5024, 128)$, 240-byte ciphertexts: optimal security within 2^{20} bytes if n and t are required to be multiples of 32.
- $(4608, 3360, 96)$, 188-byte ciphertexts: optimal security within 2^{19} bytes if n and t are required to be multiples of 32.
- $(3488, 2720, 64)$, 128-byte ciphertexts: optimal security within 2^{18} bytes if n and t are required to be multiples of 32.

Because Classic McEliece has such small ciphertexts, there would be a noticeable further savings from changing the CCA conversion to eliminate plaintext confirmation (32 bytes). However, plaintext confirmation has security advantages: for example, it stops chosen-ciphertext attacks at an earlier point than implicit rejection, making side-channel defenses easier. We have therefore decided to keep the existing CCA conversion.

It is also possible to modify parameters to further reduce ciphertext sizes at each security level by using larger keys and larger code rates. We do not plan to specify such parameter sets for now.

We already made tiny changes that affect test vectors in our August 2018 software release. Randomness is produced for key generation in a different way from before. This submission generates keys in a way that allows a secret key to be optionally compressed to a 256-bit seed; uncompressing a seed fits in low memory. This submission also implements, as a possible future proposal, an alternative key-generation method that is more complicated (and that requires larger compressed secret keys) but that is faster.