Classic McEliece: conservative code-based cryptography: modifications for round 3

10 October 2020

In the 2nd-round submission, the `f` parameter sets were documented and "implemented in this submission as a possible future proposal". In the 3rd-round submission, these are included among the official list of parameter options.

As in the 2nd-round submission, key generation is defined using various random objects, such as a random monic irreducible polynomial g over $\mathbb{F}_q$ of degree t. In the 3rd-round submission, an explicit map is specified from a random byte string to key pairs, via explicit maps from random byte strings to random objects such as g. This modularizes the tasks of reviewing (1) the correctness of implementations of this map, (2) the security of this map (i.e., the indistinguishability of g etc. from uniform), and (3) the security of the source of random bytes. This leaves open the option of specifying further maps in the future with similar security review.

The map to key pairs starts from 32 bytes of randomness, allowing a private key to be compressed to a 32-byte seed. The seed is expanded with SHAKE256. Rejection sampling in key generation is handled by deterministically mapping each seed to a new seed.

The private key now has the following format: 256 bits of the (final) seed that generates the key (not earlier rejected seeds); a 64-bit weight-32 string specifying the columns used for semi-systematic form (or a compatible constant for systematic form); the polynomial g; control bits for $(\alpha_1, \ldots, \alpha_n)$; and an n-bit string s used for implicit rejection. This leaves open the option of specifying other private-key formats in the future, such as compressed formats (the order of objects in this format is designed to allow compression via simple truncation with efficient decompression), or a list of $(\alpha_1, \ldots, \alpha_n)$ instead of control bits for environments where permutation via RAM is not a concern. Any other private-key format with efficient conversion algorithms to and from this private-key format will have the same mathematical security.