From: David Jao <djao@math.uwaterloo.ca>
Sent: Wednesday, April 17, 2019 10:57 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: [pqc-forum] ROUND 2 OFFICIAL COMMENT: SIKE

To all concerned,

It has come to our attention that the 2nd round SIKE submission contains erroneous ARM64 performance benchmarks. The error is due to some (as yet
unfixed) bug in our time measurement code, which only manifests on some devices. The existence of the error has been confirmed by hand-timing with a stopwatch.

We have re-run the ARM64 performance benchmarks on an error-free device and posted the corrected results on our web site at https://sike.org/changes.html

A corrected copy of the entire submission package is also available for download from our web site at
https://sike.org/

Generally speaking, our original (erroneous) performance numbers were 30-40% faster than reality on ARM64 for the optimized (portable) implementation, and 300-400% faster than reality for the ARM64 assembly optimized implementation.

Apologies to all for the mistake. I would be happy to discuss further with anyone who has questions. As far as I know, there are no errors in our other performance benchmarks other than ARM64.

On behalf of the SIKE team,

-David

From: Joost Renes <j.renes@cs.ru.nl>
Sent: Thursday, September 12, 2019 3:16 AM
To: pqc-comments
Cc: pqc-forum
Subject: ROUND 2 OFFICIAL COMMENT: SIKE

Dear all,

We would like to announce the results of the paper titled "Dual Isogenies and their Application to Public-key Compression for Isogeny-based Cryptography" (available at https://eprint.iacr.org/2019/499.pdf), whose results have recently been incorporated into the SIDH library v3.2 (available at https://github.com/microsoft/PQCrypto-SIDH).

Public-key compression for SIKE allows to decrease the size of public keys and ciphertext, at the cost of increased run-time. We reduce the relative overhead in run-time of including public-key compression versus excluding public-key compression (compared to the version of compression in the SIKE round-2 submission) for

* key generation from 140-153% to 61-74%,

* encapsulation from 67-90% to 38-57%,

* decapsulation from 59-65% to 34-39%.

The exact values in these ranges depend on the choice of parameters, while the sizes of public keys and ciphertexts are unchanged (compression reduces them by about 40%, e.g., a SIKEp434 public key is reduced from 330 bytes to only 196 bytes). It comes at the cost of a small increase in code complexity, and increase in table sizes. However, the sizes of the tables remain dominated by those needed for discrete logarithm computations, which we have not touched.

Kind regards,

Patrick Longa, Michael Naehrig and Joost Renes