Dear all,

The 2nd round submission of qTesla contains 3 different versions --

1. Essentially the version that was submitted to the 1st round (the Bai-Galbraith scheme with somewhat different parameters) which is now claimed secure in the ROM based on LWE and SIS (for which they incorrectly claimed security in the QROM based on LWE in the 1st round),
2. A version with much larger parameters which is currently claimed to be tightly secure based on LWE in the QROM,
3. A version of (1) that now also uses the main new idea from the original CRYSRTALS-Dilithium submission to significantly reduce the size of the public key by letting the signer send a short hint as part of the signature in lieu of the verifier needing access to a large part of the public key. This scheme is claimed to be secure in the ROM based on LWE and SIS.

We (Peter Schwabe and I) give a complete break of (3) - in particular, a signing algorithm for arbitrary messages that doesn't require the use of the secret key and is faster than the qTesla signing algorithm (which uses the secret keys). Peter's code is available at https://cryptojedi.org/qTesla-attack.tar.bz2.

The attack is not due to a bug in the qTesla implementation or any fundamental issue in the hint-generation technique from CRYSRTALS-Dilithium, but rather due to the authors of qTesla not understanding the fact that letting the signer send a hint is equivalent to letting him add an arbitrary vector of some length -- and so an adversary can do the same thing! In Dilithium, we controlled the length of this arbitrary vector (as well as the length of the signature component that it affects) and gave a security reduction from the LWE and SIS problems to the hardness of breaking the scheme. There is no such proof in qTesla. Furthermore, even for the version of qTesla without the hint (i.e. (1)), there is no reasoning in the paper as to why their parameters have the concrete hardness of SIS that they claim. One may notice that the lattice dimensions needed in qTesla are smaller than Dilithium's for the same security levels -- this is entirely due to the fact that qTesla ignores analyzing the actual hardness of the SIS problem.

Additionally, in light of the above, their statement in the change-log document that the technique for shortening the public key from (3) can be "easily extended" to their version of (2) is rather presumptuous. First, it's necessary to correctly understand the technique in order to "easily extend" it; and secondly, this was *already done* a year-and-a-half ago (before the NIST process started) in https://eprint.iacr.org/2017/916 where the scheme was called Dilithium-QROM.

So the state of affairs concerning qTesla at this point is that the parameters for (3) are completely broken, parameters for (1) have an unclear reasoning for their claimed security since the hardness of SIS is ignored, and if one wants the most compact version of (2), this was already done in https://eprint.iacr.org/2017/916

Best,

Vadim and Peter