**From:** Jan-Pieter D'Anvers <janpieter.danvers@esat.kuleuven.be>
**Sent:** Friday, January 12, 2018 10:18 AM
**To:** pqc-comments
**Cc:** pqc-forum@list.nist.gov
**Subject:** OFFICIAL COMMENT: Round2

Dear all,

I believe that the proof of the IND-CPA security of Round2 contains a flaw.

In the security proof of ROUND2.CPA-PKE, you state that the advantage of an adversary distinguishing between game G3 and G4 is less or equal then his advantage of solving the specific dGLWR problem mentioned in formula 13. However, the adversary has possibly more information in the first case (distinguishing games G3 and G4) since he is given extra information about the secret key R embedded in v (line 6-7 in Game G3 and G4). Therefore, it is possible that the advantage of the adversary in distinguishing Game G3 and G4 is actually bigger than solving the dGLWR problem. A similar problem can be found in the security proof of ROUND2.CPA-KEM.

To work around this issue, the step from Game G3 to Game G5 could be done in one step, comparing it with one similar dLWR problem (with extra samples), similar to Bos et al. [1]. However, due to the rounding, it becomes more involved when working with LWR.

For the uRound (powers of two) parameters, you can take the same route as we did in the security prove of Saber (the paper containing the proof has not been put on eprint yet). This proof proceeds from game G3 by adding two additional games G3a and G3b.

In game G3a, B is generated uniformly from R^{d/nxn}_{n,**q**}, X is calculated as <B^T R>_**q**, and v is now send with log2(t*q/p) bit coefficients. Here you can prove that the advantage of the adversary in Game G3a is at least as big in Game G3, since he can easily calculate the same values as in Game G3 by taking mod p of B and mod t of v.

In game G3b, the amount of error introduced in the coefficients of v, and the coefficients of U, is equalized. This can be done by calculating both with Rcompress_q-->max(p,t*q/p). Again, the advantage of the adversary in Game G3b is at least as big as in Game G3a, since he can easily calculate the same values as in Game G3a.

After this, you can go to Game G5 in one step analogue to Algorithm C.

Kind regards,

Jan-Pieter D'Anvers

[1] Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! Practical, Quantum-Secure Key Exchange from LWE. Cryptology ePrint Archive, Report 2016/659, 2016. https://eprint.iacr.org/2016/659

**From:** Mike Hamburg <mike@shiftleft.org>
**Sent:** Friday, January 12, 2018 7:58 PM
**To:** pqc-comments
**Cc:** pqc-forum@list.nist.gov
**Subject:** OFFICIAL COMMENT: Round2

Hello Round2 team,

Could you explain how you derived your NIST category claims?

As I understand them, the categories are roughly:

Category 1: >= 2^128/MAXDEPTH
Category 2: >= 2^128
Category 3: >= 2^192/MAXDEPTH
Category 4: >= 2^192
Category 5: >= 2^256/MAXDEPTH

where MAXDEPTH is probably between 2^40 and 2^64 for a quantum machine, and 1 for a classical machine.

Compare for example nRound2.KEM_{n=d}, as shown in Table 11. The estimated security levels against the strongest attack (hybrid) are as follows:

NIST1: 2^74
NIST2: 2^97
NIST3: 2^106
NIST4: 2^139
NIST5: 2^139

Am I misreading these tables, or misunderstanding the categories? Is the attack model here something very favorable to the attacker, so that it doesn't match NIST's model?

Thanks,
— Mike Hamburg

**From:** Martin Tomlinson <mt@post-quantum.com>
**Sent:** Friday, March 16, 2018 5:36 AM
**To:** pqc-comments
**Cc:** pqc-forum@list.nist.gov
**Subject:** OFFICIAL COMMENT: Round2
**Attachments:** signature.asc

Dear All,

All of the European Patent references appear to be scrambled in the IP statement since they do not correspond to any patents listed at the European Patent Office.

Accordingly it is impossible to determine which particular aspect of the submission is covered by one or more of the referenced patents and whether these are early stage applications or granted patents.

It will be helpful if this can be remedied promptly.

Best regards

Martin Tomlinson

**From:** Alperin-Sheriff, Jacob (Fed)
**Sent:** Friday, May 25, 2018 4:20 PM
**To:** pqc-comments
**Cc:** pqc-forum@list.nist.gov
**Subject:** Re: OFFICIAL COMMENT: Round2

Certainly this makes all of the parameters for nRound2.KEM$_{n=d}$ parameters dead because either the decryption failure rate is less than 2^{-64} for fully honest running of the algorithm or it falls to < 2^{128} classical attacks

---

**From:** "Alperin-Sheriff, Jacob (Fed)" <jacob.alperin-sheriff@nist.gov>
**Date:** Friday, May 25, 2018 at 4:12 PM
**To:** pqc-comments <pqc-comments@nist.gov>
**Cc:** "pqc-forum@list.nist.gov" <pqc-forum@list.nist.gov>
**Subject:** OFFICIAL COMMENT: Round2

Round2 Team:

Your submissions appear vulnerable to the “precomputation” reaction (CCA) attack.

To remind those of what I’m talking about, the way this sort of attack works on LWE/LWR-like schemes is

(1) Adversary receives the public key
(2) Using a large amount of computation (equivalent to more than 2^64 operations but still quite a bit less than 2^128 operations),
It identifies a large number of messages m to use in the CCA-secure scheme to generate rho (and from there to generate R) such that i_U (as used in the error analysis) will be a fair amount bigger than it would be on average. This will make decryption failure significantly more likely on these messages than on an average one.

My guess is there are somewhat more sophisticated techniques for choosing such messages such that, e.g. with even higher probability among subset of messages at least one is much higher than expected probability to cause a decryption failure (possibly choosing such messages to be closer to orthogonal to each other might help?) but I’m pretty sure even this will allow you to get a set of 2^64 messages (maximum we allow CCA queries on) such that the probability of any of these messages resulting in a decryption failure is somewhat under 2^{-64}, meaning that with high probability, decryption failures can be caused with the 2^64 required queries.

Hopefully I’ll get a more sophisticated analysis of this done next week.
—Jacob Alperin-Sheriff

**From:** Alperin-Sheriff, Jacob (Fed)
**Sent:** Wednesday, June 06, 2018 3:31 PM
**To:** pqc-comments
**Cc:** pqc-forum@list.nist.gov
**Subject:** Re: OFFICIAL COMMENT: Round2

This comment was a mistake. When I was reading through the parameters I mistakenly thought that (as for almost every other submission) the KEM was the CCA version.

Long story short, assuming the error rates given in the supporting documentation are correct (I'm a little leery about this since using prime-order cyclotomic rings should be causing a big jump in the noise unless I'm missing something …), I don't see any clear "precomputation" reaction CCA attack.

---

**From:** "Alperin-Sheriff, Jacob (Fed)" <jacob.alperin-sheriff@nist.gov>
**Date:** Friday, May 25, 2018 at 4:12 PM
**To:** pqc-comments <pqc-comments@nist.gov>
**Cc:** "pqc-forum@list.nist.gov" <pqc-forum@list.nist.gov>
**Subject:** OFFICIAL COMMENT: Round2

Round2 Team:

Your submissions appear vulnerable to the "precomputation" reaction (CCA) attack.

To remind those of what I'm talking about, the way this sort of attack works on LWE/LWR-like schemes is

(1) Adversary receives the public key
(2) Using a large amount of computation (equivalent to more than 2^64 operations but still quite a bit less than 2^128 operations),
It identifies a large number of messages m to use in the CCA-secure scheme to generate rho (and from there to generate R) such that i_U (as used in the error analysis) will be a fair amount bigger than it would be on average. This will make decryption failure significantly more likely on these messages than on an average one.

My guess is there are somewhat more sophisticated techniques for choosing such messages such that, e.g. with even higher probability among subset of messages at least one is much higher than expected probability to cause a decryption failure (possibly choosing such messages to be closer to orthogonal to each other might help?) but I'm pretty sure even this will allow you to get a set of 2^64 messages (maximum we allow CCA queries on) such that the probability of any of these messages resulting in a decryption failure is somewhat under 2^{-64}, meaning that with high probability, decryption failures can be caused with the 2^64 required queries.

Hopefully I'll get a more sophisticated analysis of this done next week.
—Jacob Alperin-Sheriff