From:	Miller, Carl A. (Fed)
To:	Perlner, Ray A. (Fed); Kelsey, John M. (Fed)
Subject:	Re: Which KEMs are affected by noise reuse, multitarget attacks.
Date:	Tuesday, October 26, 2021 10:47:37 AM

Hi Ray –

Thanks for the summary. Thinking ahead: how might we frame this in a talk? One possibility is to start with discussing standard IND-CCA security, and describe both how much security the authors are claiming, and also what we personally believe. Then we could try to list the most relevant (or most concerning) attacks that take place beyond the IND-CCA model …

-Carl

--
Carl A. Miller
Mathematician, NIST Computer Security Division
Fellow, Joint Center for Quantum Information and Computer Science (QuICS)
https://camiller.iacs.umd.edu

From: Perlner, Ray A. (Fed) <ray.perlner@nist.gov>
Date: Monday, October 25, 2021 at 1:56 PM
To: Miller, Carl A. (Fed) <carl.miller@nist.gov>, Kelsey, John M. (Fed) <john.kelsey@nist.gov>
Subject: FW: Which KEMs are affected by noise reuse, multitarget attacks.

In addition to the issues John Raised, there are a couple of issues with multitarget attacks for Classic McEliece (1 and 3 were raised by Kirk Fleming in response to some questions I asked on the forum about 2. 2 was mentioned in the CM spec in a part Dave Cooper pointed out.)

Summary for how 1,2,3 in the forwarded email affect CM:
1) The category 5 parameters for CM lose k bits of multitarget security against 2^k targets due to the use of a 256-bit seed (I think we've been saying we don't care about this for category 5 parameters.)
2) Like other Code based schemes, the other parameter sets loose 2^(k/2) bits of security due to the Decoding One out of Many (DOOM) attack. The only way to fix this is to make parameters bigger (I think we've been saying this is an annoying feature, but it's probably not worth fixing.)
3) If the same error vector is used to encapsulate for 2 different public keys, there's a pretty powerful attack. This shouldn't happen if the error vector is chosen randomly with a decent random number generator, but there are ways (like hashing the public key into a randomness seed when generating the error vector for encaps) to more definitively rule out this particular misuse scenario. Kirk Fleming considers this the most worrying of 1,2,3 for whatever that's worth.

From: Perlner, Ray A. (Fed)
Sent: Wednesday, July 21, 2021 2:43 PM
To: internal-pqc <internal-pqc@nist.gov>

Subject: Which KEMs are affected by noise reuse, multitarget attacks.

Dustin suggested I catalogue which schemes I think are affected by the attacks discussed in the forum thread with Kirk. Note, I am not infallible, please check if I missed something or am being otherwise wrong or stupid.

We discussed 3 kinds of attacks:

1. Generic multitarget attacks – these come from cases where the ciphertext is deterministically generated from a seed which is no larger than the security level, e.g. a 128-bit seed at category 1. If N ciphertexts are sent to the same public key, an attacker who sees all N ciphertexts can decrypt one of them for a cost of 2^{seedlen}/N.

SIKE,HQC, and Frodo are vulnerable to this attack at all security strength categories. In the case of SIKE, I think this attack can be fixed by simply making the seed larger. For HQC and Frodo, it's a bit more difficult, since the way these schemes use FO, the ciphertext must be recreated during decapsulation from a seed sent with the inner PKE function, and the PKE function doesn't allow a larger seed without significantly increasing parameters. I think in these cases, it's possible to fix the problem by incorporating a public salt as part of the ciphertext. We may want to suggest this fix to the affected teams.

Additionally, the category 5 parameters of Classic McEliece, Kyber, Saber, NTRU-LPRime, and BIKE are vulnerable, but we probably don't care, since even with lots of targets, the cost of the attack is still ridiculous.

2. Decoding One Out of Many (DOOM) attacks. – these are multitarget attacks that are specific to code-based schemes. As described in https://eprint.iacr.org/2011/367.pdf , for (Hamming Metric) code-based schemes, it is possible to decode one out of N ciphertexts in a code-based scheme for a cost something like 2^{single message security}/(sqrt(N)).

This affects Classic McEliece, BIKE, and HQC. There's really no way to defend against it other than increasing the parameters, which is probably not worth doing at this point. If we're feeling super doctrinaire here, I suppose we could also consider just not standardizing category 1 parameters, but a sqrt(N) speedup is probably something we can live with even at category 1.

3. Noise Reuse, for some lattice and code-based schemes, if the same noise (e.g. s and e in LWE or r and m in NTRU) is used to create ciphertexts for more than one public key, an attacker can decrypt both ciphertexts much more cheaply than the usual security level.

In the sense that there's nothing in the spec that explicitly prevents reusing noise in a dangerous way (example preventative measure: hashing the public key into the PRNG state when generating the noise), Classic McEliece, NTRU, HQC, BIKE, and Frodo are vulnerable. If we really don't trust implementers, it is possible to add a step to the random number generation part of any vulnerable spec by using the hash of a random number and the public key, rather than simply a random number. It's kind of inelegant, but it should make it harder to screw up the implementation in this

particular way, so I guess this comes down to how little we think of implementers, and how much aversion do we have to encouraging submitters to uglify their specs in the home stretch.

Best,
Ray