From: Mike Hamburg <mike@shiftleft.org>
Sent: Tuesday, March 05, 2019 8:33 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: ROUND 2 OFFICIAL COMMENT: NewHope

Hi all,

This isn't an attack on NewHope, just another note on how bad it might be if you reuse private keys in the CPA version. It came out of discussion with Mark Marson and Mélissa Rossi. Sorry if this has been discussed before, but I thought it was interesting.

Let Alice's public key be (A, As+e), and Bob's ciphertext be (B,C), so that Alice is extracting a key based on C-Bs.

Suppose that Bob chooses B to be a zero divisor. For example, in the NTT domain, b might be zero in all coefficients except one. Then there are only q=12289 different possible keys that Alice can extract.

If for some reason Alice confirms the key first, by sending eg hash(key), then Bob can just check all 12289 keys to recover that component of NTT(s). He could even do this for eg 3 nonzero coefficients at a time, at a cost of 12289^3 work per coefficient, which would mean 340-ish chosen messages to recover the key. If the key confirmation is just hash(key), he might be able to accelerate this with a rainbow table. This is stronger than other known attacks, and it's presumably possible to cut that down a little further by finishing with a lattice reduction or a combinatorial attack.

If Bob confirms the key first with SHA (which is the sane way to do it), then the attack is weaker than other known attacks, since it requires about 1024*12289/2 chosen messages. But if Bob confirms first with a polynomial MAC (eg poly1305 or GCM), and if there is some large-ish data earlier in the key exchange that Bob has control over, then Bob can compute a 12289-block message that verifies will all 12289 possible keys. Then after Alice confirms (or shares some function of the decrypted message with Bob) he can still mount the attack. So this attack wouldn't work efficiently on TLS 1.3, but it might work with some custom protocol.

This is a good reason in general that you should confirm keys with SHA and not with AEAD (or with the FO transform), and preferably as soon as possible.

It's easy to come up with countermeasures to this, such as rejecting capsules with too many coefficients equal to 0 (or q), but of course the only real countermeasure is don't reuse keys in CPA mode.

Cheers,
— Mike

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Saturday, April 06, 2019 2:56 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: signature.asc

> It's easy to come up with countermeasures to this, such as rejecting
> capsules with too many coefficients equal to 0 (or q)

The PQCrypto 2017 Gong--Zhao paper that introduced zero-divisor attacks (https://eprint.iacr.org/2016/913) also put some effort into trying to obfuscate the attacks in a way that avoids some simple countermeasures, although it's not clear how strong the obfuscation is.

> the only real countermeasure is don't reuse keys in CPA mode.

That has a more obvious effect, yes. Even more effective is to avoid standardizing any sort of "CPA mode" in the first place.

Question for the people who are proposing both IND-CPA and IND-CCA2 modes for lattice systems: Are there any publicly verifiable examples of applications where the extra cost of IND-CCA2 security is a significant part of the end user's total costs?

---Dan

From: Christopher J Peikert <cpeikert@alum.mit.edu>
Sent: Thursday, April 18, 2019 10:13 AM
To: pqc-forum; pqc-comments
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

> > the only real countermeasure is don’t reuse keys in CPA mode.
>
> That has a more obvious effect, yes. Even more effective is to avoid standardizing any sort of "CPA mode" in the first place.
>
> Question for the people who are proposing both IND-CPA and IND-CCA2 modes for lattice systems: Are there any publicly verifiable examples of applications where the extra cost of IND-CCA2 security is a significant part of the end user's total costs?

One possible example is given in Section 4.1 of https://eprint.iacr.org/2018/1037 .

It introduces "Continuous Key Agreement" (CKA), which abstracts the "double ratchet" mechanism of the Signal messaging protocol. In this context, CKA is run over an authenticated channel (provided by AEAD), so there's no need for additional authentication or validity checks. In other words, CKA aims for security against passive attacks.

The paper shows how to obtain CKA generically using any CPA-secure KEM. It then recalls Signal's optimization that saves about 2x in communication for the ElGamal KEM: a single group element plays two roles, first as a ciphertext (encapsulation), then as the sender's next public key. (See Figure 4.)

Finally, it shows that an analogous optimization is possible for CPA-secure LWE-based KEMs, thanks to their "noisy key agreement" properties.

Might the same kind of optimization be available for CCA-secure LWE-based KEMs? It's quite plausible---they include all the components of the simpler CPA-secure ones. But they also do extra (and unnecessary in this context) work of re-encrypting to check ciphertext validity. I will leave it to others to say how significant that extra work is.

In any case, this application seems like another nice advantage of the "noisy key agreement" feature of (Ring/Module-)LWE proposals.

Sincerely yours in cryptography,
Chris

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Thursday, April 18, 2019 1:18 PM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: signature.asc

I wrote:
> Question for the people who are proposing both IND-CPA and IND-CCA2
> modes for lattice systems: Are there any publicly verifiable examples
> of applications where the extra cost of IND-CCA2 security is a
> significant part of the end user's total costs?

In response, somebody advertises an application of Ring-LWE that might or might not be within scope for this standardization project, and then correctly observes that this doesn't answer my question about comparing the user's total costs to the cost of CCA2 security. ("I will leave it to others to say how significant that extra work is.")

Regarding the question at hand, let me add some further comments and then a data point regarding costs of a major application.

I wrote in https://blog.cr.yp.to/20161030-pqnist.html that "NIST should explicitly allow non-CCA2-secure single-message KEMs such as New Hope", but my support for this was explicitly conditional upon cost issues:

I prefer the simplicity of using pure encryption ... This requires
multiple-message support and CCA2 security, but my current impression
is that this robustness has only minor costs, and I wouldn't be
surprised if the New Hope team decides to move in this direction.
However, if they instead decide that CCA2 security is too expensive,
they shouldn't be rejected for targeting TLS!

New Hope did in fact decide to add support for CCA2 security. I don't see where the New Hope submission argues that the historical CPA options should be provided to users. On the contrary, the submission says

NewHope-CCA-KEM's extremely fast performance means that the cost of
this re-encryption is still quite small.

Specifically, the submission reports 220864 total Haswell cycles for NH-1024-CCA-KEM decapsulation. It's hard to see how this can be a significant cost problem compared to receiving 2208-byte ciphertexts.

I see IND-CPA and IND-CCA2 options in two other round-2 lattice submissions (LAC and Round5), plus Three Bears if that's counted as a lattice submission. I don't see where any of these submissions argue that these options should be provided to users. (Round5 cites NIST _allowing_ IND-CPA options, but that isn't the question.)

Meanwhile five round-2 lattice submissions---Frodo, Kyber, NTRU, NTRU Prime, and Saber---provide only IND-CCA2 options. (For Saber I didn't find this clearly stated in the documentation, but I see KATs only for the IND-CCA2 options.) Two of these, Kyber and (of course) NTRU Prime, state rationales for _not_ providing IND-CPA options to users:

[Kyber:] Kyber is defined as an IND-CCA2 secure KEM only. For many
applications ... active security is mandatory. However, also in use
cases (like key exchange in TLS) that do not strictly speaking
require active security, using an actively secure KEM has advantages.

Most notably ... Furthermore ... As a conclusion, we believe that the
overhead of providing CCA security is not large enough to justify
saving it and making the scheme less robust.

[NTRU Prime:] It is possible to save time, especially in
decapsulation, by abandoning protection against chosen-ciphertext
attacks. This submission intentionally avoids providing any such
options. It is not clear that the speedup is relevant to users ...
whereas there is a clear risk that providing options vulnerable to
chosen-ciphertext attacks will lead to deployment of those options in
scenarios that turn out to allow such attacks.

Obviously we can all provide IND-CPA options with faster decapsulation, but surely this should be backed by

* evidence that the speedup matters for some applications and
* an argument that the speedup outweighs the risks.

Otherwise we can and should focus on the IND-CCA2 options, simplifying analyses, benchmarks, comparisons, usage, etc.

Google, which had previously experimented with the old non-CCA2-secure version of New Hope, recently started a new experiment with something CCA2-secure (namely NTRU-HRSS), and stated that this was intentional:

CCA2-security is worthwhile, even though TLS can do without. ... CPA
vs CCA security is a subtle and dangerous distinction, and if we're
going to invest in a post-quantum primitive, better it not be fragile.

Source: https://www.imperialviolet.org/2018/12/12/cecpq2.html. Extra speed was mentioned as a bonus but evidently didn't override the CCA2 security goal.

Finally, some data regarding the costs of cryptography in context:

https://blog.cloudflare.com/how-expensive-is-crypto-anyway/

says "Cloudflare is the largest provider of TLS on the planet" and reports measurements showing that a Cloudflare server spent "just 1.8% of the CPU time" on TLS. What's even more striking is that X25519 consumed just 0.06% of the server's CPU time, even though 30% of the TLS connections exchanged keys with X25519. Some extrapolations:

* X25519 would have consumed just 0.2% (1/500) of the CPU time if it
  had been used for 100% of the connections.

* Compared to X25519, lattice systems are much bigger but typically
  use CPU time within a factor 2 (_including_ reencryption), so the
  CPU time consumed by these systems would again be negligible.

Of course it's possible that other applications are different. This brings me to the original question regarding lattice submissions: Are there any publicly verifiable examples of applications where the extra cost of IND-CCA2 security is a significant part of the end user's total costs?

---Dan

From: Rainer Urian <rainer.urian@googlemail.com>
Sent: Thursday, April 18, 2019 1:50 PM
To: D. J. Bernstein
Cc: pqc-comments; pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Performance between CCA2 and CPA is probably not an issue on high-end desktop CPUs but for sure on small (e.g. Cortex-M) chips which are used in smart cards and small IOT devices.
In smart card application you have the additional burden to make the crypto side channel and fault resistant.
This is highly required for CCA2 long-term keys but not so much for CPA ephemeral keys.

BR,
Rainer

> On Apr 18, 2019, at 7:17 PM, D. J. Bernstein <djb@cr.yp.to> wrote:
>
> I wrote:
>> Question for the people who are proposing both IND-CPA and IND-CCA2
>> modes for lattice systems: Are there any publicly verifiable examples
>> of applications where the extra cost of IND-CCA2 security is a
>> significant part of the end user's total costs?
>
> In response, somebody advertises an application of Ring-LWE that might
> or might not be within scope for this standardization project, and
> then correctly observes that this doesn't answer my question about
> comparing the user's total costs to the cost of CCA2 security. ("I
> will leave it to others to say how significant that extra work is.")
>
> Regarding the question at hand, let me add some further comments and
> then a data point regarding costs of a major application.
>
> I wrote in
> https://blog.cr.yp.to/20161030-pqnist.html that "NIST should explicitly allow non-CCA2-secure single-message KEMs such as New Hope", but my support for this was explicitly conditional upon cost issues:
>

>   I prefer the simplicity of using pure encryption ... This requires
>   multiple-message support and CCA2 security, but my current impression
>   is that this robustness has only minor costs, and I wouldn't be
>   surprised if the New Hope team decides to move in this direction.
>   However, if they instead decide that CCA2 security is too expensive,
>   they shouldn't be rejected for targeting TLS!
>
> New Hope did in fact decide to add support for CCA2 security. I don't
> see where the New Hope submission argues that the historical CPA
> options should be provided to users. On the contrary, the submission

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Thursday, April 18, 2019 5:32 PM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: signature.asc

Rainer Urian writes:
> Performance between CCA2 and CPA is probably not an issue on high-end
> desktop CPUs but for sure on small (e.g. Cortex-M) chips which are
> used in smart cards and small IOT devices.

The question regarding lattice submissions was "Are there any publicly verifiable examples of applications where the extra cost of IND-CCA2 security is a significant part of the end user's total costs?"

Here are three reasons that pointing generically at IoT devices doesn't answer the question: it

* doesn't provide a reason to think that the total cost of lattice crypto is significant compared to the total application cost;

* doesn't provide a reason to think that the cost is mainly from decapsulation time rather than from communication; and

* doesn't give us any publicly verifiable numbers.

Of course a tiny device doesn't have a quad-core 3GHz Intel CPU, but it also doesn't have a 100Mbps Ethernet connection. The numbers from

https://perso.uclouvain.be/fstandae/PUBLIS/55b.pdf

(caveat: this is already ten years old!) say that receiving a byte on an 8-bit MICAz or a 16-bit TelosB costs as much energy as, respectively,
1500 cycles of computation or 5400 cycles of computation. I'd guess that modern 32-bit Cortex-M devices have even larger ratios (and obviously they also do more per cycle), since better manufacturing rewards computation much more than communication. See, e.g., the Intel data reviewed on the top of page 46 of the NTRU Prime submission.

> In smart card application you have the additional burden to make the
> crypto side channel and fault resistant.
> This is highly required for CCA2 long-term keys but not so much for
> CPA ephemeral keys.

An application is using an IND-CCA2 lattice system. You're asking the application to switch keys as frequently as possible, to help resist physical attacks. Sounds reasonable. However, I don't see how this is relevant to the question I asked.

---Dan

From: Mike Hamburg <mike@shiftleft.org>
Sent: Thursday, April 18, 2019 8:43 PM
To: D. J. Bernstein
Cc: pqc-comments; pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Hi Dan,

> On Apr 18, 2019, at 2:32 PM, D. J. Bernstein <djb@cr.yp.to> wrote:
>
> The question regarding lattice submissions was "Are there any publicly
> verifiable examples of applications where the extra cost of IND-CCA2
> security is a significant part of the end user's total costs?"

I don't want to get involved in a debate about what level of difference is "significant". But consider that Round5 CCA and CPA versions have different parameters due to failure rate, which impacts their bandwidth in addition to CPU and memory consumption. For example, R5ND{1,3,5}KEM_5d have bandwidth {994,1639,2035} bytes, but R5ND{1,3,5}PKE_5d have bandwidth {1097,1730,2279} — a difference of {10%,6%,12%}.

A similar tradeoff holds for ThreeBears, but instead of being smaller, the CPA parameters are more secure against lattice attacks, going from {154,235,314} bits core-sieve to {168,262,351}. As with Round5, this comes at the cost of a higher failure rate (< 2^-50 instead of cryptographically negligible).

Again, a similar situation holds for BIKE, though of course that's not a lattice scheme: it's ring-LPN instead of LWE.

On the subject of performance measurements in general, I think that to make a fair performance comparison between the different KEMs, we should primarily consider the CCA versions. All the remaining submissions have one, and it's certain to be a major use case. Perhaps the following criteria would make sense for a first cut at a fair performance evaluation, at least for the lattice schemes:

* Consider the performance of the IND-CCA version of the primary recommendation at each security level. Round5 doesn't list a primary recommendation, but probably the most interesting is the most aggressive one, R5ND{1,3,5}_PKE_5d.

* Measure on Haswell (HT/TB off), Cortex-A53 (or maybe A57 or A76), and Cortex-M4 on the STM Discovery board.

* Use code which is constant-time with respect to secrets, to the extent that would be required for CCA deployment in a network-facing part. Or else benchmark decryption failures instead of successes, to measure the impact of error correction routines and of ThreeBears' optional implicit rejection code (if it survives to the next round it will adopt the state of the art CCA transform, which is reasonably likely to use implicit rejection).

* For schemes that have a SHAKE version and an AES+SHA2 version, use the SHAKE version. It's like CPA vs CCA: every scheme benefits from changing SHAKE to hardware-accelerated AES, so let's start with the SHAKE version that everyone included.

* Most schemes generate their private key from a seed. Make sure to normalize how much the implementations cache — is it just the seed, the seed and private noise, or the whole ring element / matrix (if present)?

* Different implementations have wildly different code and memory sizes. It's probably worth controlling for this by at least not manually unrolling all loops (or doing it everywhere).

* Graph the performance (bandwidth, cycles, code size, memory, energy etc) on each platform against estimated security according to your favorite estimator(s).

Note that the estimator you use matters, but it might only affect the relative position of Round5. I'm basing this on "Estimate all the LWE" on parameters from the first round (I haven't reanalyzed them), where the security with enumeration-like estimators is almost a smooth monotone function of the security with sieve-like estimators. The exceptions are Round2, NTRU Prime, PapaBear and FireSaber, but only Round2 is drastically off curve. I think at least for the first three, the issue is that hybrid attacks affect the enumeration estimates but not the sieve ones.

Maybe it would be best to ask authors to make a SUPERCOP submission including at least one version with the above criteria, once the forum can agree on the details?

Of course, you could always measure other scenarios or other platforms, but the above is easiest.

Thoughts?
— Mike

From: Rainer Urian <rainer.urian@googlemail.com>
Sent: Friday, April 19, 2019 7:28 AM
To: D. J. Bernstein
Cc: pqc-comments; pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

a concrete example could be a post quantum ICAO passport.
Current ICAO passports use a PAKE protocol (i.e. PACE) which uses an ephemeral ECDH to establish a secure channel.

Contactless communication speed is meanwhile above 6Mbps.
So we can assume that a PQ passport will have a communication speed above that.
Reading out passport data, say, 30kb , can be done in 50ms.

On the other hand, a masked Cortex M4 implementation of NewHope takes about
25.000.000 cycles for CCA2 and 500.000 cycles for CPA.
With a 100 Mhz clock, the CCA2 would need 250ms, the CPA only 5ms.

Viele Grüße / Best regards,
Rainer

> On 18. Apr 2019, at 23:32, D. J. Bernstein <djb@cr.yp.to> wrote:
>
> Rainer Urian writes:
>> Performance between CCA2 and CPA is probably not an issue on high-end
>> desktop CPUs but for sure on small (e.g. Cortex-M) chips which are
>> used in smart cards and small IOT devices.
>
> The question regarding lattice submissions was "Are there any publicly
> verifiable examples of applications where the extra cost of IND-CCA2
> security is a significant part of the end user's total costs?"
>
> Here are three reasons that pointing generically at IoT devices doesn't
> answer the question: it
>
>   * doesn't provide a reason to think that the total cost of lattice
>   crypto is significant compared to the total application cost;
>
>   * doesn't provide a reason to think that the cost is mainly from
>   decapsulation time rather than from communication; and
>
>   * doesn't give us any publicly verifiable numbers.
>
> Of course a tiny device doesn't have a quad-core 3GHz Intel CPU, but it
> also doesn't have a 100Mbps Ethernet connection. The numbers from
>
>
https://perso.uclouvain.be/fstandae/PUBLIS/55b.pdf

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Friday, April 19, 2019 10:40 AM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: signature.asc

Rainer Urian writes:
> Current ICAO passports use a PAKE protocol (i.e. PACE) which uses an
> ephemeral ECDH to establish a secure channel.
> Contactless communication speed is meanwhile above 6Mbps.
> So we can assume that a PQ passport will have a communication speed above that.
> Reading out passport data, say, 30kb , can be done in 50ms.
> On the other hand, a masked Cortex M4 implementation of NewHope takes
> about
> 25.000.000 cycles for CCA2 and 500.000 cycles for CPA.
> With a 100 Mhz clock, the CCA2 would need 250ms, the CPA only 5ms.

https://travel.state.gov/content/travel/en/passports/requirements/fees.html
says that renewing a U.S. passport costs $110. You're talking about a CPU that costs only a small percentage of this.

(For intensely competitive markets, one can try to argue that small cost differences matter. For U.S. passports, this agency doesn't have any competitors, aside from occasional black-market competitors that presumably have little effect on the overall economics. Also, the cost of a passport is only a small fraction of the cost of international travel, so there's no reason to think that small changes in passport fees will noticeably affect the number of passports obtained.)

As for time, even if 250ms is the best that can be done, it's hard to see why this matters in context. Aren't border interactions normally two orders of magnitude slower than this? (I'm ignoring the often vastly larger time spent waiting in line before the border interactions begin.) Advertisements for passport-reading machines typically say "within seconds", which sounds like much more time than is needed here.

Another passport-reading scenario outlined in

http://multimedia.3m.com/mws/media/1102183O/cs-passengershiptechnology-finalstory-aug2015.pdf

is thousands of people boarding a cruise ship "in a few hours", which suggests a budget of about 3 seconds per passport---but surely this can be, should be, and is parallelized across multiple passport readers. One of these readers costs somewhat more than $1000 according to

https://www.amazon.com/Gemalto-AT9000-Passport-Document-Reader/dp/B07DFNPMK7

but surely each reader lasts for many thousands of uses, which surely means many millions of dollars of cruises, so the costs of the passport reader are almost unnoticeable. (They're still large enough to fund someone to negotiate a bulk purchase with 3M, obviously, but this is very far from indicating that the cost of IND-CCA2 matters.)

This example looks like a great illustration of the difference between looking at cryptographic costs in isolation and looking at them as part of the total costs of the application.

---Dan

From: Oscar Garcia-Morchon <oscar.garcia-morchon@philips.com>
Sent: Friday, April 19, 2019 10:58 AM
To: pqc-forum
Cc: djb@cr.yp.to; pqc-comments
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Hi all,

Round5 proposes as NIST submissions a KEM that is an indcpa-kem and a PKE that builds on an indcca-kem.

The reason for having as a NIST KEM submission an indcpa-kem is because we improve both CPU and bandwidth compared with the indcca-kem. This is also what Mike pointed out in his email: "For example, R5ND{1,3,5}KEM_5d have bandwidth {994,1639,2035} bytes, but R5ND{1,3,5}PKE_5d have bandwidth {1097,1730,2279} — a difference of {10%,6%,12%}." We can obtain better parameters because in Round5 we optimize over a large parameter space, and we can set as targets a low enough failure rate and a high enough security level. By doing this optimization we can get to smaller key sizes.

Having as small as possible key sizes is very important. All Internet protocols will benefit of keys that are as small as possible. See for instance the text in the Round5 submission:

" An example of a protocol for which direct integration of a KEM is challenging is IKEv2 (RFC 7296). The reason is that the first message exchange in IKEv2, IKE SA INIT does not support fragmentation. If we assume Ethernet (1500 B layer 2 packets), and we use the minimal header sizes, then there is room for a 1384 B public-key/ciphertext assuming IPv4, with IPv6, this is 1364 B. Still, this misses important information that is exchanged in most real-world deployments and that further reduces the available space. Examples of such information are notification/vendor ids, a cookie that the responder could use to decide whether it might be under a DoS attack (a minimum of 12 Bytes), and an initial contact notify that tells the responder that it is the first time we are talking to it and it should clear out any stale state (8 Bytes). If NAT traversal needs to be supported, then another 56 B are required for the corresponding notify. In general, most implementations might have enough space for perhaps 1250-1300 B; smaller than that makes things easy; larger than that forces implementations to make hard decisions. All ring-based Round5 parameter sets fit in IKEv2's IKE SA INIT."

Similarly, other protocols used, e.g., in IoT have more resourced constrained links with limited data rates, small packets, and limited energy budget. Examples of those protocols are IEE 802.15.4, ZigBee, 6LoWPAN,... Many of those protocols do not have public-key solutions today because of resource constraints. The reason why in Round5 we included an IoT indcpa-kem parameter set was to try to go as small as possible and as efficient as possible so that applications relying on those protocols can also use a public-key solution. This Round5 IoT parameter set requires 736 Bytes (public-key + ciphertext). This is still a lot for many applications, but this is the smallest that we managed to get so far.

Kind regards, Oscar on behalf of the Round5 team.

On Friday, April 19, 2019 at 1:27:40 PM UTC+2, Rainer Urian wrote:

> a concrete example could be a post quantum ICAO passport.
> Current ICAO passports use a PAKE protocol (i.e. PACE) which uses
> an ephemeral ECDH to establish a secure channel.
>
> Contactless communication speed is meanwhile above 6Mbps.
> So we can assume that a PQ passport will have a communication speed above that.
> Reading out passport data, say, 30kb , can be done in 50ms.
>
> On the other hand, a masked Cortex M4 implementation of NewHope takes about
> 25.000.000 cycles for CCA2 and 500.000 cycles for CPA.
> With a 100 Mhz clock, the CCA2 would need 250ms, the CPA only 5ms.
>
> Viele Grüße / Best regards,

From: Peter Schwabe <peter@cryptojedi.org>
Sent: Friday, April 19, 2019 12:21 PM
To: Rainer Urian
Cc: D. J. Bernstein; pqc-comments; pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

'Rainer Urian' via pqc-forum <pqc-forum@list.nist.gov> wrote:

Dear Rainer, dear all,

> a concrete example could be a post quantum ICAO passport.
> Current ICAO passports use a PAKE protocol (i.e. PACE) which uses an
> ephemeral ECDH to establish a secure channel.
>
> Contactless communication speed is meanwhile above 6Mbps.
> So we can assume that a PQ passport will have a communication speed above that.
> Reading out passport data, say, 30kb , can be done in 50ms.
>
> On the other hand, a masked Cortex M4 implementation of NewHope takes
> about
> 25.000.000 cycles for CCA2 and 500.000 cycles for CPA.
> With a 100 Mhz clock, the CCA2 would need 250ms, the CPA only 5ms.

What would be the reason to apply masking to a key-exchange protocol without CCA security that can only be used with ephemeral keys? My understanding is that masking helps against DPA, but DPA is only possible if a secret is used multiple times.

All the best,

Peter

From: Alessandro Barenghi <alessandro.barenghi@polimi.it>
Sent: Friday, April 19, 2019 12:28 PM
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

On 19/04/2019 18:20, Peter Schwabe wrote:
> 'Rainer Urian' via pqc-forum <pqc-forum@list.nist.gov> wrote:
>
> Dear Rainer, dear all,
>
>> a concrete example could be a post quantum ICAO passport.
>> Current ICAO passports use a PAKE protocol (i.e. PACE) which uses an
>> ephemeral ECDH to establish a secure channel.
>>
>> Contactless communication speed is meanwhile above 6Mbps.
>> So we can assume that a PQ passport will have a communication speed above that.
>> Reading out passport data, say, 30kb , can be done in 50ms.
>>
>> On the other hand, a masked Cortex M4 implementation of NewHope takes
>> about
>> 25.000.000 cycles for CCA2 and 500.000 cycles for CPA.
>> With a 100 Mhz clock, the CCA2 would need 250ms, the CPA only 5ms.
>
> What would be the reason to apply masking to a key-exchange protocol
> without CCA security that can only be used with ephemeral keys? My
> understanding is that masking helps against DPA, but DPA is only
> possible if a secret is used multiple times.

Horizontal side channel attacks and template attacks both work with a single trace (i.e. execution of the algorithm).

Kind regards,

-- Alessandro

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Friday, April 19, 2019 1:21 PM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: signature.asc

[quote from the Round5 submission:]
> All ring-based Round5 parameter sets fit in IKEv2's IKE SA INIT.

This includes the IND-CCA2 parameter sets, right? None of the Round5 parameter sets need the minor IND-CPA squeezing (whether it's 10% or 6% or 12%) to fit into this application?

I'm puzzled to see a claim that "Having as small as possible key sizes is very important", and then various text about the size limits in this application, where the bottom line is that the size limits are large enough for every proposed parameter set to fit. In the words of the
Round5 submission, "smaller than that makes things easy". This is an example where being as small as possible _isn't_ important.

In general, it looks difficult to use protocol size cutoffs as an argument for the IND-CPA options in the round-2 lattice submissions. The
IND-CPA-vs.-IND-CCA2 size gaps that we're talking about are very small:

* LAC: 0%, if I'm reading correctly.
* New Hope: 3% or 1.4% for ciphertexts; 0% for keys.
* Round5: reportedly 10%, 6%, 12%.
* Three Bears, if that's counted as a lattice submission: 0%.

Do we have any publicly verifiable example of a protocol that simultaneously (1) has a size cutoff, (2) can't easily change the cutoff as part of a post-quantum upgrade, and (3) has this cutoff within the tiny gaps listed above?

It occurs to me that LAC and New Hope and Three Bears actually have an incentive to _avoid_ collecting data about hard-to-change size cutoffs in real protocols. Here's why: If these cutoffs are in the same ballpark as ciphertext sizes then they'll probably fall in the much larger gaps between supported ciphertext sizes, probably forcing the application to sacrifice many bits of security. This would be a much bigger change in security level than the ~10% that Mike mentioned for IND-CPA vs.
IND-CCA2 in Three Bears.

Round5 is different, since it can target many security levels. The IKEv2 example highlighted in the Round5 submission _does_ seem to force, e.g., New Hope to drop from dimension 1024 all the way down to dimension 512, which is an interesting argument against New Hope. But this example still doesn't answer the question I asked in the first place: Are there any publicly verifiable examples of applications where the extra cost of
IND-CCA2 security is a significant part of the end user's total costs?

---Dan

From: Rainer Urian <rainer.urian@googlemail.com>
Sent: Friday, April 19, 2019 2:59 PM
To: Peter Schwabe
Cc: D. J. Bernstein; pqc-comments; pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Hi Peter,

yes, I think you are right.
One would probably implement a non-masked CPA version and a masked CCA2 version only.

Viele Grüße / Best regards,
Rainer

> On 19. Apr 2019, at 18:20, Peter Schwabe <peter@cryptojedi.org> wrote:
>
> 'Rainer Urian' via pqc-forum <pqc-forum@list.nist.gov> wrote:
>
> Dear Rainer, dear all,
>
>> a concrete example could be a post quantum ICAO passport.
>> Current ICAO passports use a PAKE protocol (i.e. PACE) which uses an
>> ephemeral ECDH to establish a secure channel.
>>
>> Contactless communication speed is meanwhile above 6Mbps.
>> So we can assume that a PQ passport will have a communication speed above that.
>> Reading out passport data, say, 30kb , can be done in 50ms.
>>
>> On the other hand, a masked Cortex M4 implementation of NewHope takes
>> about
>> 25.000.000 cycles for CCA2 and 500.000 cycles for CPA.
>> With a 100 Mhz clock, the CCA2 would need 250ms, the CPA only 5ms.
>
> What would be the reason to apply masking to a key-exchange protocol
> without CCA security that can only be used with ephemeral keys? My
> understanding is that masking helps against DPA, but DPA is only
> possible if a secret is used multiple times.
>
> All the best,
>
> Peter

From: 'Rainer Urian' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Friday, April 19, 2019 4:16 PM
To: D. J. Bernstein
Cc: pqc-comments; pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

> The IKEv2
> example highlighted in the Round5 submission _does_ seem to force,
> e.g., New Hope to drop from dimension 1024 all the way down to
> dimension 512, which is an interesting argument against New Hope.

The StrongSwan IPsec/IKE implementation has a NewHope 1024 CPA extension.
Works like a charm ...

> On Apr 19, 2019, at 7:20 PM, D. J. Bernstein <djb@cr.yp.to> wrote:
>
> [quote from the Round5 submission:]
>> All ring-based Round5 parameter sets fit in IKEv2's IKE SA INIT.
>
> This includes the IND-CCA2 parameter sets, right? None of the Round5
> parameter sets need the minor IND-CPA squeezing (whether it's 10% or
> 6% or 12%) to fit into this application?
>
> I'm puzzled to see a claim that "Having as small as possible key sizes
> is very important", and then various text about the size limits in
> this application, where the bottom line is that the size limits are
> large enough for every proposed parameter set to fit. In the words of
> the
> Round5 submission, "smaller than that makes things easy". This is an
> example where being as small as possible _isn't_ important.
>
> In general, it looks difficult to use protocol size cutoffs as an
> argument for the IND-CPA options in the round-2 lattice submissions.
> The
> IND-CPA-vs.-IND-CCA2 size gaps that we're talking about are very small:
>
> * LAC: 0%, if I'm reading correctly.
> * New Hope: 3% or 1.4% for ciphertexts; 0% for keys.
> * Round5: reportedly 10%, 6%, 12%.
> * Three Bears, if that's counted as a lattice submission: 0%.
>
> Do we have any publicly verifiable example of a protocol that
> simultaneously (1) has a size cutoff, (2) can't easily change the
> cutoff as part of a post-quantum upgrade, and (3) has this cutoff
> within the tiny gaps listed above?
>

From: Peter Schwabe <peter@cryptojedi.org>
Sent: Sunday, April 21, 2019 3:55 AM
To: Alessandro Barenghi
Cc: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Alessandro Barenghi <alessandro.barenghi@polimi.it> wrote:
> On 19/04/2019 18:20, Peter Schwabe wrote:

Dear Alessandro, dear all,

> > 'Rainer Urian' via pqc-forum <pqc-forum@list.nist.gov> wrote:
> >
> > Dear Rainer, dear all,
> >
> >> a concrete example could be a post quantum ICAO passport.
> >> Current ICAO passports use a PAKE protocol (i.e. PACE) which uses
> >> an ephemeral ECDH to establish a secure channel.
> >>
> >> Contactless communication speed is meanwhile above 6Mbps.
> >> So we can assume that a PQ passport will have a communication speed above that.
> >> Reading out passport data, say, 30kb , can be done in 50ms.
> >>
> >> On the other hand, a masked Cortex M4 implementation of NewHope
> >> takes about
> >> 25.000.000 cycles for CCA2 and 500.000 cycles for CPA.
> >> With a 100 Mhz clock, the CCA2 would need 250ms, the CPA only 5ms.
> >
> > What would be the reason to apply masking to a key-exchange protocol
> > without CCA security that can only be used with ephemeral keys? My
> > understanding is that masking helps against DPA, but DPA is only
> > possible if a secret is used multiple times.
>
> Horizontal side channel attacks and template attacks both work with a
> single trace (i.e. execution of the algorithm).

Yes, absolutely. But would masking stop those attacks?

All the best,

Peter

From: Alessandro Barenghi <alessandro.barenghi@polimi.it>
Sent: Sunday, April 21, 2019 5:21 AM
To: Peter Schwabe
Cc: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

>____
> From: Peter Schwabe <peter@cryptojedi.org>
> Sent: 21 April 2019 09:55
> To: Alessandro Barenghi
> Cc: pqc-forum@list.nist.gov
> Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
>
> Alessandro Barenghi <alessandro.barenghi@polimi.it> wrote:
> > On 19/04/2019 18:20, Peter Schwabe wrote:
>
> Dear Alessandro, dear all,
>
> > > 'Rainer Urian' via pqc-forum <pqc-forum@list.nist.gov> wrote:
> > >
> > > Dear Rainer, dear all,
> > >
> > > >> a concrete example could be a post quantum ICAO passport.
> > > >> Current ICAO passports use a PAKE protocol (i.e. PACE) which uses
> > > >> an ephemeral ECDH to establish a secure channel.
> > > >>
> > > >> Contactless communication speed is meanwhile above 6Mbps.
> > > >> So we can assume that a PQ passport will have a communication speed above that.
> > > >> Reading out passport data, say, 30kb , can be done in 50ms.
> > > >>
> > > >> On the other hand, a masked Cortex M4 implementation of NewHope
> > > >> takes about
> > > >> 25.000.000 cycles for CCA2 and 500.000 cycles for CPA.
> > > >> With a 100 Mhz clock, the CCA2 would need 250ms, the CPA only 5ms.
> > >
> > > What would be the reason to apply masking to a key-exchange
> > > protocol without CCA security that can only be used with ephemeral
> > > keys? My understanding is that masking helps against DPA, but DPA
> > > is only possible if a secret is used multiple times.
> >
> > Horizontal side channel attacks and template attacks both work with
> > a single trace (i.e. execution of the algorithm).
>
> Yes, absolutely. But would masking stop those attacks?

Applying a masking countermeasure, raises the number of traces required for a template attack exponentially in the order of the masking, while providing a concrete hindrance to horizontal attacks (provided that they are taken into account when implementing the masking scheme).

In both cases, masking alone may not be sufficient to stop an attack, if it is possible for the attacker to obtain high SNR measurements of the power consumption and EM sampling of the traces of the device, but it represents a very effective component in lowering the SNR of the side channel at hand.

The typical case for a protection is a combination of masking, hiding, and possibly code morphing which in turn, is effective in stopping horizontal and template attacks.

Cheers,

Alessandro

From: Kevin Chadwick <m8il1ists@gmail.com>
Sent: Sunday, April 21, 2019 7:04 AM
To: pqc-forum@list.nist.gov
Subject: [pqc-forum] Re: OT: Sidechannel protection side effects Was: ROUND 2 OFFICIAL COMMENT: NewHope

I would like to raise something that has probably been discussed. If it has then I am unaware and apologise for time wasting.

Personally I feel like labelling side channels generally is problematic and suggests that risk analysis of each type is not really done. Perhaps that is just a side effect of it being easier to simply avoid branches etc. and label the reason as sidechannel protection and be done with it. Some may see this code as beautiful but IMO it is often horrific to read. (I have actually ripped some out of mbedtls before, obviously with much vector testing).

I feel like the cryptographic community is used to looking at any and all attacks in detail as it should and often does for a research paper or discussion.

When it affects the code however. I feel that like a product developer may decide to take measures that are outside their security risk model simply to avoid headlines of lab based attacks that don't actually apply (commercial risk model). A cryptographic library writer probably thinks he MUST and includes code that he hates because it is better than being publicly criticised and losing users/eyeballs.

Personally I think that most side channels should be dealt with in hw and don't apply in most use cases. Perhaps a smart card design should cover those risks in a special cryptographic library?

If they additionally mean that less people analyse the much more difficult to read code and that speed decreases mean less deployment of encryption.

Has the risk-benefit model of sw based side channel protections themselves been sufficiently considered, to date?

From: 赵运磊 <ylzhao@fudan.edu.cn>
Sent: Monday, April 22, 2019 12:52 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: ROUND 2 OFFICIAL COMMENT: NewHope

When using ElGamal-type KEM to replace the CPA-seucre Diffie-Hellman in TLS, there are the following problems that may cause serious security concerns:

(1) Lazy user: as the session-key is predetermined and set by one user. If it is lazy, it may re-use the same session-key for many concurrent sessions. This has already been reported with TLS based on RSA-KEM in the past.

(2) Relatively poor randomness: again, as the session-key is set by one user, the session-key may have poor randomness, compared to Diffie-Hellman type key exchange where the session-key is cooperatively generated by the two communicating parties.

If communication cost is a serious issue, the bandwidth of NewHope-KEM can actually be reduced.
The first CPA-secure KEM based on NewHope, named AKCN4:1, was developed in https://arxiv.org/abs/1611.06150 since Nov 2016. On the same parameters (same security, same error probability, etc), AKCN4:1 has smaller bandwidth than NewHope-KEM (actually AKCN4:1 and NewHope-KEM are extremely similar in mathematical structures). To further increase session-key size and decrese bandwidth and error probability simutaneously, in https://arxiv.org/abs/1611.06150, we developed AKCN-E8 (with encoding and decoding in E8 instead of D4).On the same parameters of NewHope-KEM, AKCN-E8 can have size-doubled session-key (e.g., 512-bit session-key), smaller bandwidth and/or error probability.

Best regards
Yunlei

From: Peter Pessl <peter.pessl@gmail.com>
Sent: Tuesday, April 23, 2019 7:20 AM
To: pqc-forum
Cc: peter@cryptojedi.org
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Dear Alessandro, dear all,

Alessandro Barenghi <alessandr...@polimi.it> wrote:

> Applying a masking countermeasure, raises the number of traces required
> for a template attack exponentially in the order of the masking,
> while providing a concrete hindrance to horizontal attacks (provided that
> they are taken into account when implementing the masking scheme).

In such a single-trace scenario, the argument of increasing the number of required traces shouldn't really matter. Masking might protect against horizontal (template-based or standard) DPA, like the one done in https://ia.cr/2018/687 . This then depends on the scheme, what is masked, etc., but I suspect that these are the things you meant with "taken into account". But I don't think that such a horizontal DPA is even possible for, e.g., NewHope. Also, if you can recover each share individually, then masking doesn't help at all. We did that in a previous paper https://ia.cr/2017/594 . So in my opinion, it might be better to spend the resources used for protection on better hiding instead of masking.

Cheers,
Peter
--

From: Alessandro Barenghi <alessandro.barenghi@polimi.it>
Sent: Wednesday, April 24, 2019 1:25 PM
To: Peter Pessl; pqc-forum
Cc: peter@cryptojedi.org
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

> ____
> From: Peter Pessl <peter.pessl@gmail.com>
> Sent: 23 April 2019 13:19
> To: pqc-forum
> Cc: peter@cryptojedi.org
> Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
>
> Dear Alessandro, dear all,
>
> Alessandro Barenghi <alessandr...@polimi.it> wrote:
>
> Applying a masking countermeasure, raises the number of traces
> required for a template attack exponentially in the order of the
> masking, while providing a concrete hindrance to horizontal attacks
> (provided that they are taken into account when implementing the masking scheme).
>
> In such a single-trace scenario, the argument of increasing the number
> of required traces shouldn't really matter.

If it is possible for an attacker building templates to have control on the RNG, and derive templates for the ephemeral key bits, it still has some meaning.
I know this is platform dependent, but, in case of a uC/CPU running the algorithm it shouldn't be too much of a problem to derive templates with known outputs from the RNG on the attacker-controlled clone device.

> Masking
> might protect against horizontal (template-based or standard) DPA,
> like the one done in https://ia.cr/2018/687. This then depends on the scheme, what is masked, etc., but I suspect that these are the things you meant with "taken into account".

Precisely

> But I
> don't think that such a horizontal DPA is even possible for, e.g., NewHope.
> Also, if you can recover each share individually, then masking doesn't
> help at all. We did that in a previous paper https://ia.cr/2017/594

Yes, if you are able to recover the shares individually, then performing a horizontal attack is just a matter of recombining them properly before

> So in my opinion, it might be better to spend the resources used for
> protection on better hiding instead of masking.

Good hiding helps a lot against horizontal attacks and, given the intrinsic parallelism of some primitives, it may be achieved at a reasonably low performance cost. My 2 cents are that the best trade-off point between computation overhead devoted to masking and computation time devoted to hiding will probably depend on the primitive. It would be interesting to find out that the best option is "just hiding, no masking".

Cheers,

Alessandro

From: Martin Tomlinson <mt@post-quantum.com>
Sent: Wednesday, May 29, 2019 4:56 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: Message signed with OpenPGP using GPGMail.asc

The ISARA corporation have a patent granted in 2017, US Patent 9,698,986 B1 entitled "Generating Shared Secrets For Lattice-based Cryptographic Protocols"
which seems to have some overlap with the NewHope Round 2 submission.

Claim 1 of the patent is very broad and may cover some of the other Round 2 lattice based submissions.

Maybe these are questions for NIST,
1) Does the NewHope submission (or any other lattice based submissions) need to be modified to avoid the claims of US Patent 9,698,986 ?
2) Should ISARA be approached to obtain an IP declaration regarding their intentions towards PQC submissions?

--Martin

--

From: daniel.apon <daniel.apon@nist.gov>
Sent: Wednesday, May 29, 2019 12:41 PM
To: pqc-forum
Cc: pqc-comments
Subject: Re: ROUND 2 OFFICIAL COMMENT: NewHope

Hi Martin,

I wanted to also make clear that I was speaking from a personal point of view (as opposed to NIST's official point of view, or a lawyer's point of view) in my prior response.

Thanks for understanding,
--Daniel

On Wednesday, May 29, 2019 at 10:35:59 AM UTC-4, daniel.apon wrote:

> Hi Martin,
>
> First: I am not a lawyer. Take anything I say as a layman's reading only. This should not be construed as legal advice.
>
> *1) Does the NewHope submission (or any other lattice based submissions) need to be modified to avoid the claims of US Patent 9,698,986 ?*
>
> If you examine the patent itself -- http://patft.uspto.gov/netahtml/PTO/search-bool.html search for "9,698,986 B1" -- you can see that the Detailed Description section of the patent appears to refer to New Hope as prior art. Specifically, paragraph 3 of the Detailed Description ends with "...resulting in a bandwidth savings in excess of 35% when compared with the New Hope protocol." This looks to me -- as NOT a lawyer -- as if they are primarily describing some kind of efficiency improvement to New Hope and/or RLWE-type KEMs. So, New Hope per se doesn't appear to need to be modified.
>
> *2) Should ISARA be approached to obtain an IP declaration regarding their intentions towards PQC submissions?*
>
> If you're a large enough financial target for a patent lawsuit, ask your company's patent lawyer. :-)
> We may try to check independently, but in my experience-- people tend to simply not respond to this kind of request from NIST..
> I'll update this thread if I hear anything though.
>
> --Daniel
>
> On Wednesday, May 29, 2019 at 4:56:34 AM UTC-4, Martin Tomlinson wrote:
>
>> The ISARA corporation have a patent granted in 2017, US Patent 9,698,986 B1 entitled
>> "Generating Shared Secrets For Lattice-based Cryptographic Protocols"
>> which seems to have some overlap with the NewHope Round 2 submission.

From: Mike Brown <Mike.Brown@isara.com>
Sent: Friday, May 31, 2019 12:39 PM
To: Martin Tomlinson; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Hi All,

Thanks everyone for raising this. We had the opportunity to talk to NIST and ISARA will be working together with NIST to provide a royalty-free grant to all schemes in the NIST competition. Our goal is to ensure there is no confusion or concern related to IP so we thought this would be the simplest way to achieve this. We will work with NIST on the mechanics to accomplish this.

Thanks,

Mike Brown
CTO, ISARA

On 2019-05-29, 4:57 AM, "'Martin Tomlinson' via pqc-forum" <pqc-forum@list.nist.gov> wrote:

The ISARA corporation have a patent granted in 2017, US Patent 9,698,986 B1 entitled
"Generating Shared Secrets For Lattice-based Cryptographic Protocols"
which seems to have some overlap with the NewHope Round 2 submission.

Claim 1 of the patent is very broad and may cover some of the other Round 2 lattice based submissions.

Maybe these are questions for NIST,
1) Does the NewHope submission (or any other lattice based submissions) need to be modified to avoid the claims of US Patent 9,698,986 ?
2) Should ISARA be approached to obtain an IP declaration regarding their intentions towards PQC submissions?

--Martin

--

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Saturday, June 1, 2019 2:09 PM
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: signature.asc

Mike Brown writes:
> We had the opportunity to talk to NIST and ISARA will be working
> together with NIST to provide a royalty-free grant to all schemes in
> the NIST competition.

This sounds great if it actually happens. However, I'm concerned about the following scenario:

* The _hope_ of free use of the patents leads the patents to be given lower weight in selections than they would normally be given.

* Negotiations between NIST and ISARA drag on, and eventually it turns out that NIST can't afford ISARA's buyout price.

* The selections thus end up more tilted towards ISARA's patents than they otherwise would have been.

* Users ask, quite reasonably, why patents weren't assigned a higher weight in the decision-making process.

Is there a more specific timeframe for "will be working together"?

---Dan

From: Mike Brown <Mike.Brown@isara.com>
Sent: Saturday, June 1, 2019 4:00 PM
To: D. J. Bernstein; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Just to clarify two items.

1) There is no monetary compensation involved nor have we asked for any. ISARA is providing a free, royalty-free license grant. This is to ensure no confusion on status.

2) Discussions started Friday and we will get this sorted as soon as we can.

Thanks,

Mike.

On 2019-06-01, 2:10 PM, "D. J. Bernstein" <djb@cr.yp.to> wrote:

Mike Brown writes:
> We had the opportunity to talk to NIST and ISARA will be working
> together with NIST to provide a royalty-free grant to all schemes in
> the NIST competition.

This sounds great if it actually happens. However, I'm concerned about the following scenario:

* The _hope_ of free use of the patents leads the patents to be given lower weight in selections than they would normally be given.

* Negotiations between NIST and ISARA drag on, and eventually it turns out that NIST can't afford ISARA's buyout price.

* The selections thus end up more tilted towards ISARA's patents than they otherwise would have been.

* Users ask, quite reasonably, why patents weren't assigned a higher weight in the decision-making process.

Is there a more specific timeframe for "will be working together"?

---Dan

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Saturday, June 8, 2019 4:48 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: signature.asc

Sanity checks show problems with the NewHope "provable security"
picture. My best guess is that the NewHope team will want to make the following changes: modify the "DRLWE" definition to divide the number of "samples" by n, and modify the statement of Theorem 4.4 to replace n and n with n and 2n.

I don't vouch for the correctness and applicability of the proofs after these two modifications, but with zero modifications there's a clear applicability failure (the problem assumed to be hard in the theorem statement is potentially much weaker than the analyzed problem), and with only the DRLWE modification there's a clear correctness failure.

See Section 7.5 of my latticeproofs paper for details.

---Dan

From: EL HASSANE LAAJI <e.laaji@ump.ac.ma>
Sent: Monday, July 8, 2019 11:00 PM
To: pqc-forum; pqc-comments
Subject: ROUND 2 OFFICIAL COMMENT: NewHope

Hello NewHope team;
In poly.c file of NewHope implementation.
In the "poly_mul_pointwise" function, you used the const value 3186.
How you are computed this value or what that's means.?
****

```
void poly_mul_pointwise(poly *r, const poly *a, const poly *b)
{
 int i;
 uint16_t t;
 for(i=0;i<NEWHOPE_N;i++)
 {
  t          = montgomery_reduce(3186*b->coeffs[i]); /* t is now in Montgomery domain */
  r->coeffs[i] = montgomery_reduce(a->coeffs[i] * t);  /* r->coeffs[i] is back in normal domain */
 }
}
```

****

Best Regards

From: Peter Schwabe <peter@cryptojedi.org>
Sent: Tuesday, July 9, 2019 2:51 AM
To: EL HASSANE LAAJI
Cc: pqc-forum; pqc-comments
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

EL HASSANE LAAJI <e.laaji@ump.ac.ma> wrote:
> Hello NewHope team;

Hello EL HASSANE LAAJI,

> In poly.c file of NewHope implementation.
> In the "poly_mul_pointwise" function, you used the const value 3186.
> How you are computed this value or what that's means.?
> ****************************
> void poly_mul_pointwise(poly *r, const poly *a, const poly *b) {
>   int i;
>   uint16_t t;
>   for(i=0;i<NEWHOPE_N;i++)
>   {
>   t   = montgomery_reduce(3186*b->coeffs[i]); /* t is now in
> Montgomery domain */
>   r->coeffs[i] = montgomery_reduce(a->coeffs[i] * t); /*
> r->coeffs[i] is back in normal domain */

That's 2^36 % 12289, or 2^(2*rlog) modulo q.

All the best,

Peter

From:	Thomas.Poeppelmann@infineon.com
Sent:	Wednesday, July 10, 2019 6:44 AM
To:	djb@cr.yp.to; pqc-comments
Cc:	pqc-forum; Thomas.Poeppelmann@infineon.com
Subject:	RE: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Dear all, dear Dan,

Indeed that is an inconsistency in the specification, thanks for bringing it to our attention, Dan. We've fixed the notation in the DRLWE definition to make it clear how it counts samples, and in Theorem 4.4 we've replaced the sample counts in the advantage statements with 1 and 2 samples, respectively. An updated version of the specification with these changes fixed has been posted at https://newhopecrypto.org/
(direct link: https://newhopecrypto.org/data/NewHope_2019_07_10.pdf)

On a side note, we have also fixed several typos and details in the failure analysis (Section D) of the original NewHope paper to which we reference in the NIST submission. The updated paper can be found at https://eprint.iacr.org/2015/1092. We are thankful to Christian Berghoff for thoroughly reporting and helping us to correct several mistakes here.

Thomas (on behalf of the NewHope team)

-----Original Message-----
From: D. J. Bernstein <djb@cr.yp.to>
Sent: Samstag, 8. Juni 2019 22:48
To: pqc-comments@nist.gov
Cc: pqc-forum@list.nist.gov
Subject: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Sanity checks show problems with the NewHope "provable security"
picture. My best guess is that the NewHope team will want to make the following changes: modify the "DRLWE" definition to divide the number of "samples" by n, and modify the statement of Theorem 4.4 to replace n and n with n and 2n.

I don't vouch for the correctness and applicability of the proofs after these two modifications, but with zero modifications there's a clear applicability failure (the problem assumed to be hard in the theorem statement is potentially much weaker than the analyzed problem), and with only the DRLWE modification there's a clear correctness failure.

See Section 7.5 of my latticeproofs paper for details.

From:	David G <dgotrik@gmail.com>
Sent:	Wednesday, September 4, 2019 12:25 AM
To:	pqc-forum
Cc:	Thomas.Poeppelmann@infineon.com
Subject:	Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Hi Thomas,

I wanted to reach out to you and the NewHope team before consulting the PQC group regarding this paper "A Complete and Optimized Key Mismatch Attack on NIST Candidate NewHope" [https://eprint.iacr.org/2019/435.pdf]

"Then, inspired by Ding et al.'s key mismatch attack, we propose an efficient strategy which with a probability of 96.88% succeeds in recovering all the coefficients in the secret key. Experiments show that our proposed method is very efficient, which completes the attack in about 137.56 ms using the NewHope parameters"

What are your thoughts on this paper, and does the newer version of your paper address this? I searched through the PQC group postings and haven't found any references to this paper specifically. Could you address this? If you'd like to keep the discussions public, I can post my question on the forums too. I see that there was some minor discussion on Twitter, but so far I haven't been able to find anything else regarding this.

Kind regards,
David Gotrik

On Wednesday, July 10, 2019 at 4:44:25 AM UTC-6, Thomas.P...@infineon.com wrote:

> Dear all, dear Dan,
>
> Indeed that is an inconsistency in the specification, thanks for bringing it to our attention, Dan. We've fixed the notation in the DRLWE definition to make it clear how it counts samples, and in Theorem 4.4 we've replaced the sample counts in the advantage statements with 1 and 2 samples, respectively. An updated version of the specification with these changes fixed has been posted at https://newhopecrypto.org (direct link: https://newhopecrypto.org/data/NewHope_2019_07_10.pdf).
>
> On a side note, we have also fixed several typos and details in the failure analysis (Section D) of the original NewHope paper to which we reference in the NIST submission. The updated paper can be found at https://eprint.iacr.org/2015/1092. We are thankful to Christian Berghoff for thoroughly reporting and helping us to correct several mistakes here.
>
> Thomas (on behalf of the NewHope team)
>
> -----Original Message-----
> From: D. J. Bernstein <d...@cr.yp.to>
> Sent: Samstag, 8. Juni 2019 22:48
> To: pqc-co...@nist.gov
> Cc: pqc-...@list.nist.gov
> Subject: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
>
> Sanity checks show problems with the NewHope "provable security"
> picture. My best guess is that the NewHope team will want to make the following changes: modify the "DRLWE" definition to divide the number of "samples" by n, and modify the statement of Theorem 4.4 to replace n and n with n and 2n.

From: Leo Ducas <leo.ducas1@gmail.com>
Sent: Wednesday, September 4, 2019 11:33 AM
To: pqc-forum
Cc: Thomas.Poeppelmann@infineon.com
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Dear David,

the result of this paper (https://eprint.iacr.org/2019/435)
is a refinement in a long line of research, including (non-exhaustively):

https://eprint.iacr.org/2016/085
https://eprint.iacr.org/2016/1176
https://eprint.iacr.org/2019/075

This line studies active attacks (CCA) on schemes (or versions of schemes) designed to only be passively secure (CPA). While it does vindicate the importance of CCA security for non-ephemeral uses of such schemes, it does *not* affect the security claims of either the CPA or the CCA version of NewHope.

In the paper you cited, the CPA vs. CCA ambiguity is resolved in the last Section 5:

`` It is worth noting that the NewHope KEM submitted to NIST is CPA secure, which is then transformed into CCA-secure using Fujisaki-Okamoto transformation. Therefore, the proposed key mismatch attack does not harm the NewHope designers' security goals. ''

Best regards
-- Leo

Le mercredi 4 septembre 2019 06:24:48 UTC+2, David G a écrit :

Hi Thomas,

I wanted to reach out to you and the NewHope team before consulting the PQC group regarding this paper "A Complete and Optimized Key Mismatch Attack on NIST Candidate NewHope" [https://eprint.iacr.org/2019/435.pdf]

"Then, inspired by Ding et al.'s key mismatch attack, we propose an efficient strategy which with a probability of 96.88% succeeds in recovering all the coefficients in the secret key. Experiments show that our proposed method is very efficient, which completes the attack in about 137.56 ms using the NewHope parameters"

What are your thoughts on this paper, and does the newer version of your paper address this? I searched through the PQC group postings and haven't found any references to this paper specifically. Could you address this? If you'd like to keep the discussions public, I can post my question on the forums too. I see that there was some minor discussion on Twitter, but so far I haven't been able to find anything else regarding this.

Kind regards,
David Gotrik

On Wednesday, July 10, 2019 at 4:44:25 AM UTC-6, Thomas.P...@infineon.com wrote:

From: EL HASSANE LAAJI <e.laaji@ump.ac.ma>
Sent: Tuesday, December 10, 2019 5:28 PM
To: pqc-forum; pqc-comments
Subject: NewHope second round comment (about NTT functions)

Hi Newhope Team;

I have two questions.

1- First question.

Why the inverse of NTT function poly_invntt(&ehat); does not return the exact normal form?
I tray the ***poly_ntt(&ehat)*** and ***poly_invntt(&ehat)*** functions, but that not return the exact normal form of ehat polynomial

poly_invntt(poly_ntt(&ehat))≠ ehat

```
……………….
 poly_sample(&ehat, noiseseed, 1);
  printf("\n 1- NORMAL FORM  of polynomial ehat :  \n");
                            for(int i=0; i<NEWHOPE_N;i++) printf("%d .",ehat.coeffs[i]);
  poly_ntt(&ehat);
  printf("\n 2- NTT FORM  of polynomial ehat by using poly_ntt(&ehat) function:  \n");
                            for(int i=0; i<NEWHOPE_N;i++) printf("%d .",ehat.coeffs[i]);
  poly_invntt(&ehat);
  printf("\n 3 NORMAL FORM of polynomial ehat after using invNTT   \n");
              for(int i=0; i<NEWHOPE_N;i++) printf("%d .",ehat.coeffs[i]);
……………….
```

The result
1- NORMAL FORM of polynomial ehat :
12287 .12286 .12288 .12290 .12290 .12290 .12291 .12295 .12290 .12291 .12287 .12292 .12289…
2- NTT FORM of polynomial ehat by using poly_ntt(&ehat) function:
11420 .7102 .1471 .1301 .10492 .1871 .8348 .3085 .7169 .11779 .4186 .2455 .2326 .8704 .11898…
3 NORMAL FORM of polynomial ehat after using poly_invntt(&ehat) function
12287 .12291 .12290 .12285 .12291 .12292 .12287 .12290 .12288 .12290 .12287 .0 .12286 .12288…

2- Second question.

In function
*int crypto_kem_enc(unsigned char *ct, unsigned char *ss, const unsigned char *pk)*

```
{
1.  unsigned char buf[2*NEWHOPE_SYMBYTES];
 2.  randombytes(buf,NEWHOPE_SYMBYTES);
 3.  shake256(buf,2*NEWHOPE_SYMBYTES,buf,NEWHOPE_SYMBYTES
 4.  cpapke_enc(ct, buf, pk, buf+NEWHOPE_SYMBYTES);
 5.  shake256(ss, NEWHOPE_SYMBYTES, buf, NEWHOPE_SYMBYTES);
  6. return 0;
}
```

In **line 5** the 3rd argument (buf) in **shake256(.)** has no relation with **(ct) !**
Can you explain to me these problems?
Best regards.

From: EL HASSANE LAAJI <e.laaji@ump.ac.ma>
Sent: Sunday, December 15, 2019 10:48 AM
To: pqc-comments; pqc-forum
Subject: Re: ROUND 2 OFFICIAL COMMENT: NewHope

re Hi NewHope Team;
Another remark. In ***void cpapke_keypair(unsigned char *pk, unsigned char *sk)*** function, you use the pointwise multiplication function ***poly_mul_pointwise(&ahat_shat, &shat, &ahat)***; but you did not transform **ahat** polynomial to NTT form.

```
........
gen_a(&ahat, publicseed);
 poly_sample(&shat, noiseseed, 0);
 poly_ntt(&shat);
 poly_sample(&ehat, noiseseed, 1);
 poly_ntt(&ehat);
 poly_mul_pointwise(&ahat_shat, &shat, &ahat);
.........
```

If there are mistakes, how the shared secret **(ss)** is the same for both ***int crypto_kem_enc(unsigned char *ct, unsigned char *ss, const unsigned char *pk)*** *and* ***int crypto_kem_dec(unsigned char *ss, const unsigned char *ct, const unsigned char *sk)*** functions.

Best Regards.

Le sam. 14 déc. 2019 à 00:35, EL HASSANE LAAJI <e.laaji@ump.ac.ma> a écrit :

> Hi Newhope Team;
> Why the inverse of NTT function poly_invntt(&ehat); does not return the exact normal form?
> I tray the ***poly_ntt(&ehat)*** and ***poly_invntt(&ehat)*** functions, but that not return the exact normal form of ehat polynomial
>
> ***poly_invntt(poly_ntt(&ehat))≠ ehat***
>
> ```
> ……………….
>  poly_sample(&ehat, noiseseed, 1);
>   printf("\n 1- NORMAL FORM  of polynomial ehat :  \n");
>                            for(int i=0; i<NEWHOPE_N;i++) printf("%d .",ehat.coeffs[i]);
>   poly_ntt(&ehat);
>   printf("\n 2- NTT FORM  of polynomial ehat by using poly_ntt(&ehat) function:  \n");
>                            for(int i=0; i<NEWHOPE_N;i++) printf("%d .",ehat.coeffs[i]);
>   poly_invntt(&ehat);
>   printf("\n 3 NORMAL FORM of polynomial ehat after using invNTT   \n");
>               for(int i=0; i<NEWHOPE_N;i++) printf("%d .",ehat.coeffs[i]);
> ……………….
> ```
>
> **The result**
> **1- NORMAL FORM of polynomial ehat :**
> 12287 .12286 .12288 .12290 .12290 .12290 .12291 .12295 .12290 .12291 .12287 .12292 .12289…
> 2- **NTT FORM of polynomial ehat by using poly_ntt(&ehat) function:**
> 11420 .7102 .1471 .1301 .10492 .1871 .8348 .3085 .7169 .11779 .4186 .2455 .2326 .8704 .11898…
> **3 NORMAL FORM of polynomial ehat after using poly_invntt(&ehat) function**
> 12287 .12291 .12290 .12285 .12291 .12292 .12287 .12290 .12288 .12290 .12287 .0 .12286 .12288…
>
> Can you explain to me what is the problem?
> Best regards.

From: Peter Schwabe <peter@cryptojedi.org>
Sent: Tuesday, December 17, 2019 8:23 AM
To: EL HASSANE LAAJI
Cc: pqc-forum; pqc-comments
Subject: Re: [pqc-forum] NewHope second round comment (about NTT functions)

EL HASSANE LAAJI <e.laaji@ump.ac.ma> wrote:
> Hi Newhope Team;

Hi EL HASSANE LAAJI,

> I have two questions.
>
> 1- First question.
>
> Why the inverse of NTT function poly_invntt(&ehat); does not return
> the exact normal form?

Input to the NTT is assumed to be in bit-reversed order. This works, because all inputs to the NTT are noise polynomials, so permuting the coefficients does not really gain anything and costs performance. As a consequence however, poly_ntt and poly_invntt are not inverses of each other; you'd need to include a bit-reversal at the beginning of poly_ntt to ensure that they are.

> [...]

> *2- **Second question.*
>
> In function
>
> *int crypto_kem_enc(unsigned char *ct, unsigned char *ss, const
> unsigned char *pk)*
>
> {
>
> 1. unsigned char buf[2*NEWHOPE_SYMBYTES];
>
> 2. randombytes(buf,NEWHOPE_SYMBYTES);
>
> 3. shake256(buf,2*NEWHOPE_SYMBYTES,buf,NEWHOPE_SYMBYTES
>
> 4. cpapke_enc(ct, buf, pk, buf+NEWHOPE_SYMBYTES);
>
>
> * 5. shake256(ss, NEWHOPE_SYMBYTES, buf, NEWHOPE_SYMBYTES); *
>
> 6. return 0;
>
> }
>
> In *line 5* the 3rd argument (buf) in *shake256(.) *has no relation
> with*

> (ct) !*
>
> Can you explain to me these problems?

Why would this be a problem? The ciphertext (ct) is sent to the communication partner. The shared key (ss) is derived from the plaintext of the encryption (hashed together with some other values). The communicating partner decrypts to obtain the same plaintext and is then able to derive the same shared key.

All the best,

The NewHope team

From: Peter Schwabe <peter@cryptojedi.org>
Sent: Tuesday, December 17, 2019 8:27 AM
To: EL HASSANE LAAJI
Cc: pqc-comments; pqc-forum
Subject: Re: [pqc-forum] Re: ROUND 2 OFFICIAL COMMENT: NewHope

EL HASSANE LAAJI <e.laaji@ump.ac.ma> wrote:
> re Hi NewHope Team;

Dear EL HASSANE LAAJI,

> Another remark. In *void* *cpapke_keypair(unsigned char *pk, unsigned
> char
> *sk)* function, you use the pointwise multiplication function*
> poly_mul_pointwise(&ahat_shat, &shat, &ahat)*; but you did not
> transform *ahat* polynomial to NTT form.
> ....
> gen_a(&ahat, publicseed);
>   poly_sample(&shat, noiseseed, 0);
>   poly_ntt(&shat);
>   poly_sample(&ehat, noiseseed, 1);
>   poly_ntt(&ehat);
>   poly_mul_pointwise(&ahat_shat, &shat, &ahat); ....
> If there are mistakes, how the shared secret *(ss*) is the same for
> both *int crypto_kem_enc(unsigned char *ct, unsigned char *ss, const
> unsigned char
> *pk) **and int crypto_kem_dec(unsigned char *ss, const unsigned char
> *ct, const unsigned char *sk)* functions.

The NTT transforms uniformly random values into uniformly random values.
So, we *could* sample and then transform to get ahat, but what we do is to sample and simply assume that the sampled polynomial is already in NTT domain. See the paragraph "Definition of GenA" on page 9 of the Newhope Specification Document: https://newhopecrypto.org/data/NewHope_2019_07_10.pdf

All the best,

Peter

From: EL HASSANE LAAJI <e.laaji@ump.ac.ma>
Sent: Friday, January 3, 2020 5:25 PM
To: pqc-comments; pqc-forum
Subject: ROUND 2 OFFICIAL COMMENT: NewHope " shared secret"

Hi NewHope Team.
I have remark about Shared Secret:
In " int crypto_kem_enc(unsigned char *ct, unsigned char ***ss_e**, const unsigned char *pk)" encryption function and in "int crypto_kem_dec(unsigned char ***ss_d,** const unsigned char *ct, const unsigned char *sk)" decryption function.
When the decryption is good we find the shared secret keys **ss_e** and **ss_d** are equal.

for(int **i=0;i<CRYPTO_BYTES;++i)**printf("%0X.",**ss_e**[i]);
E2.D1.49.E4.53.F2.27.B4.D9.67.86.19.4C.36.5E.4B.43.21.E0.12.77.A3.77.64.D3.84.C3.10.3D.4F.97.8F.

for(int **i=0;i<CRYPTO_BYTES;++i**)printf("%0X.",**ss_d**[i]);
E2.D1.49.E4.53.F2.27.B4.D9.67.86.19.4C.36.5E.4B.43.21.E0.12.77.A3.77.64.D3.84.C3.10.3D.4F.97.8F.

But if the decryption fails, we find the exact shared secret key in the second half of **ss_d :**
for(int **i=CRYPTO_BYTES;i<2*CRYPTO_BYTES;++i**)printf("%0X.",**ss_d**[i]);
E2.D1.49.E4.53.F2.27.B4.D9.67.86.19.4C.36.5E.4B.43.21.E0.12.77.A3.77.64.D3.84.C3.10.3D.4F.97.8F.
Even without the secret key **shat, (**we make the line of the secret key in comment in decryption function) as below::

```
void cpapke_dec(unsigned char *m,
           const unsigned char *c,
           const unsigned char *sk)
{
  poly vprime, uhat, tmp, shat;
  //poly_frombytes(&shat, sk);
  decode_c(&uhat, &vprime, c);
  poly_mul_pointwise(&tmp, &shat, &uhat);
  poly_invntt(&tmp);
  poly_sub(&tmp, &tmp, &vprime);
  poly_tomsg(m, &tmp);
}
```

I think the vulnerability is in KEM functions!
Best regards.

From: Lukas Prokop <lukas.prokop@iaik.tugraz.at>
Sent: Monday, January 20, 2020 5:47 AM
To: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope " shared secret"

Hi EL HASSANE LAAJI,

On 03.01.20 23:25, EL HASSANE LAAJI wrote:
> I have remark about Shared Secret:
> In " int crypto_kem_enc(unsigned char *ct, unsigned char **ss_e*,
> const unsigned char *pk)" encryption function and in "int
> crypto_kem_dec(unsigned char **ss_d,* const unsigned char *ct, const
> unsigned char *sk)" decryption function.
> *When* the decryption is good we find the shared secret keys *ss_e*
> and *ss_d *are equal.
>
> for(int* i=0;i<CRYPTO_BYTES;++i)*printf("%0X.",*ss_e*[i]);
> E2.D1.49.E4.53.F2.27.B4.D9.67.86.19.4C.36.5E.4B.43.21.E0.12.77.A3.77.64.D3.84.C3.10.3D.4F.97.8F.
>
> for(int *i=0;i<CRYPTO_BYTES;++i*)printf("%0X.",*ss_d*[i]);
> E2.D1.49.E4.53.F2.27.B4.D9.67.86.19.4C.36.5E.4B.43.21.E0.12.77.A3.77.64.D3.84.C3.10.3D.4F.97.8F.
>
> *But *if the decryption fails, we find the exact shared secret key in
> the second half of *ss_d :*
> for(int*
> i=CRYPTO_BYTES;i<2*CRYPTO_BYTES;++i*)printf("%0X.",*ss_d*[i]);
> E2.D1.49.E4.53.F2.27.B4.D9.67.86.19.4C.36.5E.4B.43.21.E0.12.77.A3.77.64.D3.84.C3.10.3D.4F.97.8F.
> *E*ven without the secret key* shat, (*we make the line of the secret
> key in comment in decryption function) as below::
> void cpapke_dec(unsigned char *m,
> const unsigned char *c,
> const unsigned char *sk) {
>
> *poly vprime, uhat, tmp, shat; //poly_frombytes(&shat, sk);
> decode_c(&uhat, &vprime, c); poly_mul_pointwise(&tmp, &shat, &uhat);
> poly_invntt(&tmp); poly_sub(&tmp, &tmp, &vprime); poly_tomsg(m,
> &tmp);* }

To the best of my knowledge, you create a buffer overflow. The first argument of `crypto_kem_dec` has `CRYPTO_BYTES` bytes, as documented.
You access elements at index ≥ CRYPTO_BYTES.

Can you provide public key (pk), secret key (sk), and the exact location of insertion of your printf-loops? Otherwise this example is irreproducible and cannot be analyzed any further.

all the best,
Lukas

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Thursday, July 23, 2020 5:33 AM
To: pqc-comments
Cc: pqc-forum
Subject: ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: signature.asc

https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf has a new provable-security argument concluding that "the security of NewHope is never better than that of KYBER". (This is almost half of NIST's text in that document regarding NewHope.)

I don't believe the argument. I'm filing this comment to request that NIST spell out the argument in more detail for public review. (I'm also skeptical regarding the conclusion of the argument, but that's a more subtle issue.)

Section 7.3 of https://cr.yp.to/papers.html#latticeproofs observes that "proofs do not reach all the way from one of the target KEMs to another, or even from one of the target PKEs to another". As far as I know, this remains the situation today, and is not changed by the subsequent paper that NIST cites as part of its argument (nor do I see where that paper claims to change the situation).

I understand that NewHope is not a round-3 candidate, and that NIST could switch to other justifications for eliminating NewHope, but the problems with this argument seem relevant to round 3 both procedurally (where exactly did the public have an opportunity to review and correct NIST's provable-security claim?) and content-wise (the security of MLWE and Kyber will continue to be compared to other round-3 candidates, and in any case should not be exaggerated).

---Dan

From: Damien Stehlé <damien.stehle@gmail.com>
Sent: Friday, July 24, 2020 11:58 AM
To: pqc-forum; pqc-comments
Cc: Leo Ducas; Vadim Lyubashevsky
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Dear Dan, everyone,

As far as we understand, the situation is as follows:

1. There is a worst-case to worst-case "everything-preserving" (i.e., noise, samples over Z_q, total dimension over Z) reduction from Ring-LWE instances to Module-LWE ones, 2. There is an average-case to average-case reduction from Ring-LWE to Module-LWE that is noise/total dimension preserving, but not sample-preserving, 3. There is no known noise-preserving reduction of any kind going the other way around (i.e., from Module-LWE to Ring-LWE), 4. There is no known "everything-preserving" average-case to average-case reduction in either direction.

To sum up, there is no known reduction between Kyber and NewHope, due to Item 4. However, Items 1 - 3 make it seem unlikely (at least to
us) that Module-LWE could be easier in practice for "natural" (e.g., uniform in bounded intervals) distributions that are not covered by the reductions. This was one of the reasons why Module-LWE was chosen for Kyber.

Best regards,
Damien, Léo, Vadim

On Thu, 23 Jul 2020 at 11:32, D. J. Bernstein <djb@cr.yp.to> wrote:
>

From: Christopher J Peikert <cpeikert@alum.mit.edu>
Sent: Friday, July 24, 2020 12:07 PM
To: Damien Stehlé
Cc: Leo Ducas; Vadim Lyubashevsky; pqc-comments; pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

On Fri, Jul 24, 2020 at 11:57 AM Damien Stehlé <damien.stehle@gmail.com> wrote:

> Dear Dan, everyone,
>
> As far as we understand, the situation is as follows: ...
>
> 2. There is an average-case to average-case reduction from Ring-LWE
> to Module-LWE that is noise/total dimension preserving, but not
> sample-preserving,

Well, the reduction is even "sample preserving," in that it maps one Ring-LWE sample to one Module-LWE sample.

However, Kyber reveals more Module-LWE samples than NewHope reveals in Ring-LWE samples. This is the why there is no known reduction between (the pre-FO versions of) Kyber and NewHope.

Sincerely yours in cryptography,
Chris

> 3. There is no known noise-preserving reduction of any kind going
> the other way around (i.e., from Module-LWE to Ring-LWE),
> 4. There is no known "everything-preserving" average-case to
> average-case reduction in either direction.
>
> To sum up, there is no known reduction between Kyber and NewHope,
> due to Item 4. However, Items 1 - 3 make it seem unlikely (at least to
> us) that Module-LWE could be easier in practice for "natural" (e.g.,
> uniform in bounded intervals) distributions that are not covered by the
> reductions. This was one of the reasons why Module-LWE was chosen
> for Kyber.
>
> Best regards,
> Damien, Léo, Vadim
>
> On Thu, 23 Jul 2020 at 11:32, D. J. Bernstein <djb@cr.yp.to> wrote:
> >
> > https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf has a new
> > provable-security argument concluding that "the security of NewHope is
> > never better than that of KYBER". (This is almost half of NIST's text in
> > that document regarding NewHope.)
> >
> > I don't believe the argument. I'm filing this comment to request that
> > NIST spell out the argument in more detail for public review. (I'm also

From:	daniel.apon <daniel.apon@nist.gov>
Sent:	Friday, July 24, 2020 2:08 PM
To:	pqc-forum
Cc:	cpei...@alum.mit.edu; Leo Ducas; vadim....@gmail.com; pqc-comments; pqc-forum; damien...@gmail.com
Subject:	Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Hi Dan, and all,

It takes us a little extra time to formulate a full response as an entire group, so this email may be slightly repetitive of what others have said just recently in this thread. In any case--

As you point out, most of the text in https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf regarding NewHope references KYBER. The discussion there primarily aimed to explain the reasoning leading to the statement "NIST developed a slight but clear preference for KYBER and for low-rank MLWE schemes over RLWE schemes," particularly given (i) the selection of KYBER as a finalist in the 3rd Round, and (ii) the overwhelming similarities between NewHope and KYBER in general.

The underlying technical argument can be (and has been) stated many ways, but we found the peer-reviewed and publicly-published paper "Algebraically Structured LWE, Revisited" (TCC 2019) to have the technically-tightest and most general argument about the relationship between RLWE and MLWE.

This paper's proceedings version has been accessible to the public on the publisher's website at https://link.springer.com/chapter/10.1007%2F978-3-030-36030-6_1 since November 22, 2019, and it has been accessible to the public in pre-print form at https://eprint.iacr.org/2019/878 since August 1, 2019. All versions of the pre-print (August 1, 2019; September 21, 2019; and November 15, 2019) are accessible to the public at https://eprint.iacr.org/eprint-bin/versions.pl?entry=2019/878. For the sake of discussion, we may use the November 15, 2019 version of the paper, currently found at https://eprint.iacr.org/2019/878.pdf.

The technical matter relevant to the NewHope vs KYBER decision point is found in Section 6 "Reduction from O'-LWE to O-LWE^k" beginning on Page 15. Our interpretation is that for the case of power-of-2 cyclotomics, there is a reduction from Ring-LWE to Module-LWE that's nearly perfect. We list some features of this reduction:

-- The reduction runs in linear time.
-- The reduction scales up security by ring-dimension n times module-rank k.
-- The reduction is modulus-preserving.
-- The reduction is "almost" sample-preserving. (*)
-- The reduction is error distribution and error magnitude preserving in the "typical setting," where the coordinates of the errors are chosen i.i.d.

(*) The Module-LWE secret in KYBER is exposed to more samples (say, 3 or 4) than the Ring-LWE secret in NewHope (just 1). This appears to be inherent in the natural constructions, as the typical plain LWE cryptosystem 'gives away' a linear number of samples.

Due to this above caveat, there is not a formal proof that an attack on KYBER yields an attack on NewHope (ceteris paribus regarding exact concrete parameterization choices). However, there is also no proof or even clear reason to believe that this difference helps an attacker. (It would, for example, be surprising to conclude that "typical" plain LWE cryptosystems are fundamentally easier to attack than low-rank Module-LWE cryptosystems, primarily because the plain

LWE cryptosystem releases more samples.)

We do note, separately, that we have meaningful, fairly tight reductions from power-of-2 cyclotomic Ring-LWE (under mildly differing parameters) to both NewHope and KYBER. So the conclusion drawn is that -- from the high-level perspective -- NewHope and KYBER are roughly equally good in terms of security. (We also note KYBER has a reduction from Module-LWE, but there is no known reduction from Module-LWE to NewHope.)

Therefore, in terms of Ring-LWE vs Module-LWE (and NewHope vs KYBER), we assess (with a slight preference) that the more conservative, more security-conscious choice is the less algebraically-structured option of Module-LWE. Since security is Job #1, our decision aligned with what we assess to be the higher-security option.

Finally, we note that the above discussion (modulo the sample-count caveat mentioned) primarily relates to our assessment of the relative security of IND-CPA NewHope and IND-CPA KYBER. There is the separate, outstanding question of whether the CPA-to-CCA transformation affects this picture. Speaking "informally," there doesn't yet seem to be a significant reason to believe so, although it is certainly possible.

--The NIST PQC team

On Friday, July 24, 2020 at 12:07:55 PM UTC-4 cpei...@alum.mit.edu wrote:

> On Fri, Jul 24, 2020 at 11:57 AM Damien Stehlé <damien...@gmail.com> wrote:
>
>> Dear Dan, everyone,
>>
>> As far as we understand, the situation is as follows: ...
>>
>> 2. There is an average-case to average-case reduction from Ring-LWE
>> to Module-LWE that is noise/total dimension preserving, but not
>> sample-preserving,
>
> Well, the reduction is even “sample preserving,” in that it maps one Ring-LWE sample to one Module-LWE sample.
>
> However, Kyber reveals more Module-LWE samples than NewHope reveals in Ring-LWE samples. This is the why there is no known reduction between (the pre-FO versions of) Kyber and NewHope.
>
> Sincerely yours in cryptography,
> Chris
>
>> 3. There is no known noise-preserving reduction of any kind going
>> the other way around (i.e., from Module-LWE to Ring-LWE),
>> 4. There is no known "everything-preserving" average-case to
>> average-case reduction in either direction.
>>
>> To sum up, there is no known reduction between Kyber and NewHope,
>> due to Item 4. However, Items 1 - 3 make it seem unlikely (at least to
>> us) that Module-LWE could be easier in practice for "natural" (e.g.,
>> uniform in bounded intervals) distributions that are not covered by the
>> reductions. This was one of the reasons why Module-LWE was chosen
>> for Kyber.
>>
>> Best regards,
>> Damien, Léo, Vadim

From: Damien Stehlé <damien.stehle@gmail.com>
Sent: Saturday, July 25, 2020 4:36 AM
To: Christopher J Peikert
Cc: Leo Ducas; Vadim Lyubashevsky; pqc-comments; pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope

Le ven. 24 juil. 2020 à 18:07, Christopher J Peikert <cpeikert@alum.mit.edu> a écrit :

> On Fri, Jul 24, 2020 at 11:57 AM Damien Stehlé <damien.stehle@gmail.com> wrote:
>
>> Dear Dan, everyone,
>>
>> As far as we understand, the situation is as follows: ...
>>
>> 2. There is an average-case to average-case reduction from Ring-LWE
>> to Module-LWE that is noise/total dimension preserving, but not
>> sample-preserving,
>
> Well, the reduction is even "sample preserving," in that it maps one Ring-LWE sample to one Module-LWE sample.
>
> However, Kyber reveals more Module-LWE samples than NewHope reveals in Ring-LWE samples. This is the why there is no known reduction between (the pre-FO versions of) Kyber and NewHope.
>
> Sincerely yours in cryptography,
> Chris

Dear Chris, all,

Well, "sample" is a bit overloaded here.

If one counts RingLWE samples rather than the number of ModuleLWE samples that a RingLWE sample contains, then the worst-case to worst-case reduction (Item 1) increases the number of samples.
One RingLWE sample contains a number of (correlated) ModuleLWE samples that is equal to the ratio of field degrees.

Best regards
Damien

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Saturday, July 25, 2020 4:37 AM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: NewHope
Attachments: signature.asc

Damien Stehlé writes:
> 4. There is no known "everything-preserving" average-case to
> average-case reduction in either direction.
> To sum up, there is no known reduction between Kyber and NewHope, due
> to Item 4.

Actually, Item 4 understates the obstacles to a proof. Even if, hypothetically, a reduction can simultaneously

* handle the average case,
* match the number of samples, and
* exactly preserve errors,

this _still_ wouldn't give a security proof saying that Kyber is at least as strong as NewHope. It wouldn't work

* even if the KEMs are simplified down to the PKEs,
* even if the PKEs are simplified to ignore ciphertext compression,
* even if the PKEs are simplified to ignore error correction, and
* even if the number of samples in the cryptosystems is adjusted to
  match what the reductions need,

contrary to what readers would expect from your message (and from Peikert's message, and from NIST's message).

The big extra problem is NewHope's wide error distribution. This is not something obscure: the wide error distribution plays a clear role in the NewHope security analysis (see Section 4.1.1), and in the picture of lattice security proofs more generally. The wide error distribution also plays a clear role in the discrepancy between

* RLWE being more efficient than rank-2/3/4/... MLWE but
* NewHope being less efficient than Kyber.

The changes from NewHope to Kyber include switching to a much narrower error distribution, adding up just 4 bits mod 3329 instead of 16 bits mod 12289. The "modulus switching" proof technique is far too noisy to compensate for this. The Kyber documentation doesn't claim otherwise.
Surely you agree that this change destroys any hope of a Kyber>=NewHope proof---it's not just that there's no "known" reduction.

It's awfully tempting at this point to

* quote the portion of the Kyber documentation describing "very low
  noise" as a much larger "threat" than something else, and to

* comment on the relative vulnerability levels of Kyber and NewHope

to _known_ hybrid attacks.

But diving into this tangent would be a distraction from the topic at hand. Someone disputing a claim that X is provably at least as secure as Y shouldn't be asked to demonstrate that X is less secure than Y. The people claiming proofs---in this case, NIST---are responsible for avoiding exaggerations of proofs in the first place. It isn't okay to gloss over average-case issues, it isn't okay to gloss over sample issues, it isn't okay to gloss over compression etc., and it isn't okay to gloss over changes in the error distribution.

The bigger picture is that lattice-based cryptography is constantly deceiving potential users regarding what has been proven. The worst part of this is the bait and switch between

* constant advertising of worst-case-to-average-case theorems and

* switching, for deployment proposals, to efficient cryptosystems for which the theorems (at best) say security >= 2^tiny.

Users are only occasionally warned that there are two different types of lattice cryptosystems, the theorem type and the efficient type. The switch from "theorem" to "efficient" includes, most importantly,

(1) taking distributions too narrow for the theorems and
(2) taking dimensions too small for the theorems.

Effect #1 is illustrated by NewHope disappearing in favor of Kyber--- the supposed upgrade from RLWE to MLWE doesn't compensate for the loss of error distribution. Effect #1 is also illustrated by your own proposal of wide NTRU distributions (used in a round-1 NISTPQC submission, exactly for its worst-case-to-average-case story) disappearing in favor of narrow distributions.

Effect #2 is illustrated by _all_ of the NISTPQC lattice submissions.
This fact tends to be buried behind details omitted from asymptotic theorem statements, but this fact is adequately documented in the literature (starting with https://eprint.iacr.org/2016/360.pdf) and isn't subject to any scientific dispute.

I'm not saying that giving up the theorems is a bad idea. (The theorems, when applicable, provide far less security assurance than commonly believed; consume resources that would be better applied to other risk-reduction techniques; and seem incompatible with NIST's pursuit of
performance.) But we have to warn users that worst-case-to-average-case theorems simply do not apply to lattice systems proposed for deployment.

With this background, it's particularly striking to see NIST stating that NTRU "lacks a formal worst-case-to-average-case reduction". This doesn't make any sense as a comment unless NIST believes, incorrectly, that some of the lattice submissions _have_ a worst-case-to-average-case reduction. An incorrect belief in a worst-case-to-average-case reduction for (say) Kyber would necessarily include

(1) missing the error-distribution obstacle to Kyber proofs and
(2) missing the dimension obstacle to Kyber proofs.

Missing the error-distribution obstacle, and downplaying "minor" issues such as the number of samples, would also make a Kyber>=NewHope proof sound plausible. Sure enough, this imaginary security proof, concluding that "the security of NewHope is never better than that of KYBER", is the centerpiece of NIST's NewHope commentary.

NIST promised more transparency after Dual EC. I don't understand why NIST keeps soliciting _private_ NISTPQC input rather than asking for the whole evaluation to be done in public. (I also said this on the record before round 3 was announced.) This isn't just an NSA issue; anyone who has served on program committees sees that some scientists use privacy as a shield for exaggeration. I don't see NISTPQC procedures to compensate for overconfidence and confirmation bias; I don't see where NIST asked for public feedback regarding these NTRU-vs.-something and NewHope-vs.-Kyber provable-security claims before the assessments appeared in the latest report; and I don't see how similar NIST errors in round 3 are going to be corrected before they're used for decisions.

---Dan