From: Peter Schwabe <peter@cryptojedi.org> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
CC: authors@pq-crystals.org
Subject: [pqc-forum] Kyber decisions, part 2: FO transform
Date: Saturday, December 03, 2022 08:11:21 PM ET

Dear all,

This is the second mail about possible tweaks to Kyber as part of the standardization. Kyber as specified in round-3 (and also previous rounds) uses a tweaked Fujisaki-Okamoto transform to build a CCA-secure KEM from a CPA-secure PKE. Specifically, Kyber hashes the hash of the public key into the random coins and the shared key and Kyber hashes the hash of the ciphertext into the shared key. The reasons for those tweaks are the following:

* Hashing the (hash of the) public key into the final key makes the KEM "contributory", i.e., the shared key depends on inputs from both parties;

* hashing the (hash of the) public key into the random coins gives protection against multi-target attacks exploiting decryption failures; and

* hashing the (hash of the) ciphertext into the shared key ensures that this shared key depends on the full transcript.

Through the course of the NIST PQC project, multiple papers considered the FO transform and also the tweaked version used in Kyber. The question for standardization is if the results of these papers should be incorporated into Kyber, or not:

1.) https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Feprint.iacr.org%2F2021%2F1351&data=05%7C01%7Cyi-kai.liu%40nist.gov%7Cf1a966a359f74f40152708dad594728e%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638057130811457870%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=YEosYLvcO9wVOiQEvOdl5iRtg%2FgucPhoJ%2FOd7A7V8YM%3D&reserved=0 shows that as a protection against

multi-target failure attacks it is not necessary to make random coins dependent on the full public key. It is sufficient to hash in a prefix of the public key, if that prefix has sufficiently high min-entropy. We are not aware of any formal definition of a KEM being "contributory", but intuitively also for this property using such a prefix would be sufficient. Using prefix(pk) instead of H(pk) would require fewer Keccak permutations in Kyber and thus speed up encapsulation. Should the Kyber standard use prefix(pk) rather than H(pk)?

2.) Hashing the (hash of the) ciphertext into the final shared key does not help at all with any formal security property or with proofs. On the contrary, hashing the hash of the ciphertext into the final key complicates QROM proofs as pointed out in https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Feprint.iacr.org%2F2021%2F708.pdf&data=05%7C01%7Cyi-kai.liu%40nist.gov%7Cf1a966a359f74f40152708dad594728e%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638057130811457870%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=hv6BgcSzhTKWd0kLCRdwTSqtujZfnZItYDUwYhxZB1U%3D&reserved=0. Removing this tweak simplifies proofs and speeds up encapsulation. Note that decapsulation will still need to compute a hash over the full ciphertext for implicit rejection and, to avoid timing side channels, needs to do so in every decapsulation, not just after a decryption failure. So, there won't be any performance gain in decapsulation. Should the Kyber standard drop hashing the hash of the ciphertext into the shared key?

The obvious disadvantage with both possible changes is that such changes at such a late stage require very careful evaluation. We may have missed some non-standard property that Kyber achieves with these tweaks, but does not without. Also, the modifications are not completely orthogonal to potential modifications of symmetric crypto (see the previous mail), because they require changes to the hashing inside the FO transform with possible consequences for domain separation.

Again, we're looking forward to hear what everybody thinks!

All the best,

The Kyber team

--

From: Markku-Juhani O. Saarinen <mjos.crypto@gmail.com> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
CC: Peter Schwabe <peter@cryptojedi.org>, aut...@pq-crystals.org <authors@pq-crystals.org>
Subject: [pqc-forum] Re: Kyber decisions, part 2: FO transform
Date: Sunday, December 04, 2022 11:28:44 PM ET

On Sunday, December 4, 2022 at 2:11:00 AM UTC+1 Peter Schwabe wrote:

> (...) The
> question for standardization is if the results of these papers should be
> incorporated into Kyber, or not:
>
> 1.) https://eprint.iacr.org/2021/1351 shows that as a protection against
> multi-target failure attacks it is not necessary to make random
> coins dependent on the full public key. It is sufficient to hash in
> a prefix of the public key, if that prefix has sufficiently high
> min-entropy. We are not aware of any formal definition of a KEM
> being "contributory", but intuitively also for this property using
> such a prefix would be sufficient. Using prefix(pk) instead of H(pk)
> would require fewer Keccak permutations in Kyber and thus speed up
> encapsulation. Should the Kyber standard use prefix(pk) rather than
> H(pk)?

Hi Peter,

Note that performance advantage depends on the length of the "prefix" somewhat:

Kyber.CCAKEM.Enc, line 3. (K, r) :=G(m || H(pk))

In a masked side-channel secure implementation, the function H is a "plain" Keccak (currently SHA3-256), while G is a "secure" Keccak (masked SHA3-512). This is because of variable classification; H(pk) is a public value, but m is literally a secret, as are K and r outputs.

So if the prefix is too long to force additional permutation in "G", then this would cause significant performance degradation in a side-channel secure implementation.

> 2.) Hashing the (hash of the) ciphertext into the final shared key does
> not help at all with any formal security property or with proofs. On
> the contrary, hashing the hash of the ciphertext into the final key

> complicates QROM proofs as pointed out in https://eprint.iacr.org/2021/708.pdf. Removing this tweak simplifies proofs and speeds up encapsulation. Note that decapsulation will still need to compute a hash over the full ciphertext for implicit rejection and, to avoid timing side channels, needs to do so in every decapsulation, not just after a decryption failure. So, there won't be any performance gain in decapsulation. Should the Kyber standard drop hashing the hash of the ciphertext into the shared key?

Can you elaborate with a fuller pseudocode of the proposal? Are you proposing replacing [Kyber.CCAKEM.Enc, line 5] "K := KDF(k || H(c))" with "K := KDF(k)" or just using "k" (the first half of "G(m || H(pk))" or "G(m || prefix)" as the shared secret?

Implicit rejection itself does not use H(c); [Kyber.CCAKEM.Dec, line 7] compares c itself with re-encrypted c'. Does the decapsulation then need to compute KDF(K) (or whatever the encapsulation does) *and* some KDF(z || H(c)) -- and then do a conditional selection between these two results?

(Clearly, if one simplifies the implicit rejection from [Kyber.CCAKEM.Dec, line 10] "K := KDF(z || H(c))" to simply "K := KDF(z)", then every invalid ciphertext query would yield the same K value and would hence be distinguishable and break Fujisaki-Okamoto.)

Cheers,

- markku

Dr. Markku-Juhani O. Saarinen <mjos@iki.fi>

From: Peter Schwabe <peter@cryptojedi.org> via pqc-forum@list.nist.gov
To: Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
CC: pqc-forum <pqc-forum@list.nist.gov>, Peter Schwabe <peter@cryptojedi.org>, aut...@pq-crystals.org <authors@pq-crystals.org>
Subject: [pqc-forum] Re: Kyber decisions, part 2: FO transform
Date: Monday, December 05, 2022 03:27:59 AM ET

"Markku-Juhani O. Saarinen" <mjos.crypto@gmail.com> wrote:

> On Sunday, December 4, 2022 at 2:11:00 AM UTC+1 Peter Schwabe wrote:
>
> > (...) The
> >>
> > question for standardization is if the results of these papers should be
> >> incorporated into Kyber, or not:
> >>
> >> 1.) https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Feprint.iacr.org%2F2021%2F1351&data=05%7C01%7Cyi-kai.liu%40nist.gov%7Cc3685439719b4f37a79a08dad69a9cd2%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638058256794333039%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=4FUJ3syXYHyI9nk4szDG0ssnWNdYcj9JKDrpR59%2FmrQ%3D&reserved=0 shows that as a protection against
> >> multi-target failure attacks it is not necessary to make random
> >> coins dependent on the full public key. It is sufficient to hash in
> >> a prefix of the public key, if that prefix has sufficiently high
> >> min-entropy. We are not aware of any formal definition of a KEM
> >> being "contributory", but intuitively also for this property using
> >> such a prefix would be sufficient. Using prefix(pk) instead of H(pk)
> >> would require fewer Keccak permutations in Kyber and thus speed up
> >> encapsulation. Should the Kyber standard use prefix(pk) rather than
> >> H(pk)?
> >>
> > >
> > >
> > Hi Again,

Hi Markku, hi all,

> A bit more detail is required on this too. The public key in Kyber is

> defined as
>
> pk := (Encode_12 (t̂ mod q) || ρ)
>
> Hence the prefix(pk) would be chopped from the NTT-transformed t value as
> it is stored before the seed "rho" in the public key.
>
> However, the paper cited above states, *"For the 32-byte prefix ID(pk) of
> the public key in Kyber and Saber one can take the seed ρ that is already
> of size 32 bytes and uniformly random in these schemes."*
>
> So are you, in fact, suggesting changing the replacement of H(pk) with
> "rho" ?
>
> If not (and I don't see why this would be the case) —— as a task,
> cryptanalysis of the min-entropy and other properties of "t" appears to be
> basically in the domain of symmetric cryptanalysis. I think the team should
> propose some specific length.

Details to be discussed, but I believe that the "prefix" function should include all of rho and some bytes of t. Just using rho is fine if really every user generates their own rho uniformly at random. I would include, say, 32 bytes of t, just in case that at some point some application uses the same rho for multiple keys. Also, throwing in 32 bytes of t is essentially free, because we'd still just need one block of Keccak.

All the best,

Peter

From: Peter Schwabe <peter@cryptojedi.org> via pqc-forum@list.nist.gov
To: Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
CC: pqc-forum <pqc-forum@list.nist.gov>, Peter Schwabe <peter@cryptojedi.org>, aut...@pq-crystals.org <authors@pq-crystals.org>
Subject: [pqc-forum] Re: Kyber decisions, part 2: FO transform
Date: Monday, December 05, 2022 10:14:39 AM ET

"Markku-Juhani O. Saarinen" <mjos.crypto@gmail.com> wrote:

> On Monday, December 5, 2022 at 9:27:52 AM UTC+1 Peter Schwabe wrote:
>
> >
> > Details to be discussed, but I believe that the "prefix" function should
> > include all of rho and some bytes of t. Just using rho is fine if really
> > every user generates their own rho uniformly at random. I would include,
> > say, 32 bytes of t, just in case that at some point some application
> > uses the same rho for multiple keys. Also, throwing in 32 bytes of t is
> > essentially free, because we'd still just need one block of Keccak.
> >
>
> Hi Peter,

Hi Markku, hi all,

> To avoid confusion with "prefix" let's define a function "trunc(x, n)"
> which refers to the first n bytes of x (bytes 0 to n-1 inclusive).
>
> If you're concretely proposing replacing [Kyber.CCAKEM.Enc(pk), line 4]
> "G(m || H(pk))" with "G(m || rho || trunc(t, 32))" then that would imply a
> 32+32+32 = 96-byte input to G, which is a SHA3-512. That would be two
> blocks since the rate of SHA3-512 is only (1600-2*512)/8 = 72 bytes.
> Furthermore, those two permutations would be with the more expensive masked
> Keccak since this is a key derivation function (involving secrets).

Yes, you're absolutely right, I answered too quickly and didn't take the
32 bytes of m into account.

> It would also expand [Kyber.CCAKEM.Dec, line 2] "h" from "H(pk)" to "rho ||

> trunc(t, 32)", which is 32+32 = 64 bytes. The secret key grows by 32 bytes,
> right?

Well, for the secret key there's all kinds of possible tradeoffs between
size and time. As pk is anyway already part of the secret key, I would
probably not additionally store rho || trunc(t, 32).

> It would fit in a single permutation if one changes G to be 64 bytes
> extracted from SHAKE-256 (rate 136 bytes). This appears to be just as
> secure from a symmetric cryptanalysis perspective (in this particular
> case), albeit conflicts with the domain separation with KDF (which is also
> SHAKE-256.)

Agreed.

> Looking quickly, the contents of trunc(t, 32) are some 21 first
> coefficients of the first polynomial (there are "k" polynomials in "t") in
> NTT domain, which seems indistinguishable from "uniform mod q numbers put
> through Encode12 to generate 12-bit numbers". This is a different
> distribution from actual uniform bits, so the min-entropy isn't quite
> optimal. Almost but not quite. However, I do understand the "just in case"
> justification for this part.

Yes. With SHAKE-256 it would also be possible to throw in a few more
bytes of t without going for a second block.

> ps. We'd still need details of the other matter; implicit rejection without
> H(c); how to use "z."

We'll work out the details of a concrete proposal and send this soon.

All the best,

Peter

--

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Kyber decisions, part 2: FO transform
Date: Monday, December 05, 2022 03:59:46 PM ET
Attachments: smime.p7m

> We are not aware of any formal definition of a KEM
> being "contributory", but intuitively also for this property using
> such a prefix would be sufficient.

I'm having trouble seeing how the above statement is in line with the following quotes from the latest version of the submission:

> These hashes are not necessary for the security reduction (see Section 4), but they add robustness. Specifically, the final shared key output by Kyber.CCAKEM depends on the full view of exchanged messages (public key and ciphertext), which means that the KEM is contributory and safe to use in authenticated key exchanges without additional hashing of context. ...
>
> As also mentioned in Section 1.5, hashing the public key and the ciphertext is not required for CCA security, but is instead done to make the function more robust and directly usable in applications that require the shared key to depend on the entire transcript. Our rationale is that because the basic operations comprising Kyber are extremely fast, we can afford to pay a time penalty and make the default version of Kyber as robust and misuse-resilient as possible.

https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf

To clarify, is the Kyber team now abandoning the "full view", "entire transcript", and "as robust and misuse-resilient as possible" properties advertised in the Kyber submission? And is the Kyber team saying that the "contributory" property advertised in the Kyber submission never had a clear meaning?

If the intent is to downgrade the list of advertised cryptosystem

features in this way, then these changes should be stated explicitly. If the intent is instead to claim that the advertised cryptosystem features continue to apply to prefix hashing, then there obviously needs to be an explanation of, e.g., what "entire transcript" is supposed to mean.

——D. J. Bernstein

P.S. Third time trying to send this message. I wonder how many people have simply given up.

From: Peter Schwabe <peter@cryptojedi.org> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Kyber decisions, part 2: FO transform
Date: Tuesday, December 06, 2022 12:12:58 AM ET

"D. J. Bernstein" <djb@cr.yp.to> wrote:

Dear Dan, dear all,

> > We are not aware of any formal definition of a KEM
> > being "contributory", but intuitively also for this property using
> > such a prefix would be sufficient.
>
> I'm having trouble seeing how the above statement is in line with the
> following quotes from the latest version of the submission:
>
>   These hashes are not necessary for the security reduction (see
>   Section 4), but they add robustness. Specifically, the final shared
>   key output by Kyber.CCAKEM depends on the full view of exchanged
>   messages (public key and ciphertext), which means that the KEM is
>   contributory and safe to use in authenticated key exchanges without
>   additional hashing of context. ...
>
>   As also mentioned in Section 1.5, hashing the public key and the
>   ciphertext is not required for CCA security, but is instead done to
>   make the function more robust and directly usable in applications
>   that require the shared key to depend on the entire transcript. Our
>   rationale is that because the basic operations comprising Kyber are
>   extremely fast, we can afford to pay a time penalty and make the
>   default version of Kyber as robust and misuse-resilient as possible.
>

> https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fpq-crystals.org%2Fkyber%2Fdata%2Fkyber-specification-round3-20210804.pdf&data=05%7C01%7Cyi-kai.liu%40nist.gov%7Cca41a780023241e9266808dad7488814%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638059003782149457%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=kQmeKjTS4CNHVunUIUULsTdTR5Nd29mJFVpyDneLKGg%3D&reserved=0
>
> To clarify, is the Kyber team now abandoning the "full view", "entire
> transcript", and "as robust and misuse-resilient as possible" properties
> advertised in the Kyber submission? And is the Kyber team saying that
> the "contributory" property advertised in the Kyber submission never had
> a clear meaning?

> If the intent is to downgrade the list of advertised cryptosystem
> features in this way, then these changes should be stated explicitly. If
> the intent is instead to claim that the advertised cryptosystem features
> continue to apply to prefix hashing, then there obviously needs to be an
> explanation of, e.g., what "entire transcript" is supposed to mean.

We are not abandoning anything. We are summarizing discussions that we are aware of and that we think are useful to have within the whole community before the Kyber standard is written.

To answer your questions:

- If there is a clear consensus in the community that H(c) (or c) should not be hashed into the shared key, then indeed, this would be dropping the "full view", and "entire transcript" properties advertised in the Kyber submission.

- Also if there is clear consensus in the community that Kyber should hash prefix(pk) instead of H(pk) into the final shared key, this would, for all useful definitions of "prefix", mean that the "full view" and "entire transcript" properties would be dropped.

- As far as I'm aware, the adjective "contributory" for a KEM typically

> means that the final shared key depends on input from both parties
> with sufficiently large min-entropy so that no party can by themselves
> choose the final shared key. I am not aware of any paper formalizing
> this notion.

All the best,

Peter

From: Varun Maram <msvr81@gmail.com> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
CC: Peter Schwabe <peter@cryptojedi.org>, xagawa@gmail.com
Subject: Re: [pqc-forum] Kyber decisions, part 2: FO transform
Date: Wednesday, January 18, 2023 01:23:58 AM ET

Dear all,
We recently posted a preprint titled "Post-Quantum Anonymity of Kyber" [https://eprint.iacr.org/2022/1696.pdf] and we believe some of our results are quite relevant to this discussion. The main focus of our paper is to formally establish Kyber's anonymity (or, ANO-CCA security) in the QROM, with the hope of providing confidence in the scheme's compatibility with appropriate privacy-preserving applications.

Along the way, we also provide a proof of IND-CCA security for Kyber in the QROM with concrete bounds -- taking into account the **specific variant** of the FO transform used by the scheme (i.e., with the nested hashes of public-key and ciphertext in key-derivation). In summary, we believe Kyber's FO transform does not need any tweaks at-least from a provable security point-of-view in the QROM.

A technical overview of our results is presented below. Any feedback on our work is greatly appreciated!

Best,
Varun and Keita

The FO variant used by Kyber can be seen as applying a "wrapper" around a well-known FO transform **with explicit rejection** in the literature known as *FO^\bot_m* [see https://eprint.iacr.org/2017/604.pdf e.g.]. It is relatively straightforward to prove IND-CCA security of KEMs obtained from this wrapper transform -- including Kyber -- in the QROM while relying on IND-CCA security of *FO^\bot_m*-based KEMs. In recent work, Don et. al. [https://eprint.iacr.org/2021/280.pdf] and Hövelmanns et. al. [https://eprint.iacr.org/2022/365.pdf] also proved concrete IND-CCA security of the latter KEMs in the QROM. However, their analyses assume certain properties of the underlying passively-secure (i.e., IND-CPA secure) PKE scheme, and these properties are not well-studied with respect to Kyber. Their security bounds are also

non-tight when compared to the state-of-the-art bounds for KEMs in the QROM obtained from the **implicitly rejecting** counterpart of *FO^\bot_m*, namely *FO^\not\bot_m*.

Starting with the above observation, in our work, we prove the concrete IND-CCA security of KEMs obtained from the above wrapper transform in the QROM while relying on IND-CCA security of **implicitly rejecting***FO^\not\bot_m*-based KEMs. This means that the tight bounds we have for the latter KEMs in the literature -- e.g., the bounds shown by Bindel et. al. [https://eprint.iacr.org/2019/590.pdf] and Kuchta et. al. [https://eprint.iacr.org/2021/454.pdf] -- also apply to Kyber in the QROM. Note that the aforementioned tight analyses assume the underlying passively-secure PKE scheme satisfies a notion of "injectivity"; however in recent work, Ding et. al. [https://link.springer.com/chapter/10.1007/978-3-031-22301-3_17] showed that Kyber indeed satisfies such a notion.

There was also a suggestion [https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/8k3MhD_5stk/m/TWGKtuL4BgAJ] to prove IND-CCA security of Kyber in the QROM using quantum indifferentiability results of Zhandry [https://eprint.iacr.org/2018/276.pdf]. However as we argue in the paper, in addition to getting (slightly) tighter bounds from our approach, at a conceptual level, we rely on the well-known "One-Way To Hiding (OW2H) lemma" [https://eprint.iacr.org/2018/904.pdf] proof technique in our QROM analysis. It is worth pointing out that Unruh [https://eprint.iacr.org/2020/962.pdf] showed a framework for formally verifying post-quantum security proofs that involve applications of the OW2H lemma; hence, we believe this should make our IND-CCA security proof for Kyber also amenable to formal verification.

On Tuesday, December 6, 2022 at 10:42:22 AM UTC+5:30 Peter Schwabe wrote:

> "D. J. Bernstein" <d...@cr.yp.to> wrote:
>
> Dear Dan, dear all,
>
> > > We are not aware of any formal definition of a KEM
> > > being "contributory", but intuitively also for this property using
> > > such a prefix would be sufficient.
> >
> > I'm having trouble seeing how the above statement is in line with the
> > following quotes from the latest version of the submission:

> These hashes are not necessary for the security reduction (see
> Section 4), but they add robustness. Specifically, the final shared
> key output by Kyber.CCAKEM depends on the full view of exchanged
> messages (public key and ciphertext), which means that the KEM is
> contributory and safe to use in authenticated key exchanges without
> additional hashing of context. ...
>
> As also mentioned in Section 1.5, hashing the public key and the
> ciphertext is not required for CCA security, but is instead done to
> make the function more robust and directly usable in applications
> that require the shared key to depend on the entire transcript. Our
> rationale is that because the basic operations comprising Kyber are
> extremely fast, we can afford to pay a time penalty and make the
> default version of Kyber as robust and misuse-resilient as possible.
>
> https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf
>
> To clarify, is the Kyber team now abandoning the "full view", "entire
> transcript", and "as robust and misuse-resilient as possible" properties
> advertised in the Kyber submission? And is the Kyber team saying that
> the "contributory" property advertised in the Kyber submission never had
> a clear meaning?

> If the intent is to downgrade the list of advertised cryptosystem
> features in this way, then these changes should be stated explicitly. If
> the intent is instead to claim that the advertised cryptosystem features
> continue to apply to prefix hashing, then there obviously needs to be an
> explanation of, e.g., what "entire transcript" is supposed to mean.

We are not abandoning anything. We are summarizing discussions that we are aware of and that we think are useful to have within the whole community before the Kyber standard is written.

To answer your questions:

- If there is a clear consensus in the community that H(c) (or c) should

> not be hashed into the shared key, then indeed, this would be dropping the "full view", and "entire transcript" properties advertised in the Kyber submission.
>
> - Also if there is clear consensus in the community that Kyber should hash prefix(pk) instead of H(pk) into the final shared key, this would, for all useful definitions of "prefix", mean that the "full view" and "entire transcript" properties would be dropped.
>
> - As far as I'm aware, the adjective "contributory" for a KEM typically means that the final shared key depends on input from both parties with sufficiently large min-entropy so that no party can by themselves choose the final shared key. I am not aware of any paper formalizing this notion.
>
> All the best,
>
> Peter

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Kyber decisions, part 2: FO transform
Date: Wednesday, January 18, 2023 06:52:32 AM ET
Attachments: smime.p7m

Varun Maram writes:

> Along the way, we also provide a proof of IND-CCA security for Kyber
> in the QROM with concrete bounds

To clarify, am I correctly understanding that

* the proof assumes IND-CPA security for the underlying PKE, and

* the concrete bounds for Kyber-1024 (or Kyber-512) apply only if the attacker is limited to 2^82 (or 2^67) hash calls and doesn't have an IND-CPA attack with success chance above 2^-166 (or 2^-136)?

Is there literature claiming that such low-probability IND-CPA attacks are infeasible against Kyber, and quantifying the claimed security level for comparison to NIST's security requirements?

I realize that integrating depth limits into the proof should change the numbers somewhat, but in any case applying the proof strategy seems to require an analysis of low-probability attacks, and I don't see why those should be assumed to cost as much as high-probability attacks. My recent paper https://cr.yp.to/papers.html#lprrr shows one way to exploit this difference for asymptotically faster attacks assuming standard heuristics. A concrete analysis is harder than an asymptotic analysis, but I'd presume that Kyber loses security from the same effect.

——D. J. Bernstein

From: Varun Maram <msvr81@gmail.com> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
CC: D. J. Bernstein <djb@cr.yp.to>, pqc-...@list.nist.gov <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Kyber decisions, part 2: FO transform
Date: Thursday, January 19, 2023 11:37:08 PM ET

Dear Dan,

> * the proof assumes IND-CPA security for the underlying PKE, and
> * the concrete bounds for Kyber-1024 (or Kyber-512) apply only if the attacker is limited to 2^82 (or 2^67) hash calls and doesn't have an IND-CPA attack with success chance above 2^-166 (or 2^-136)?

Yes, that is correct. But one thing worth mentioning is that when it comes to the IND-CPA advantage, as per the current bounds stated in our paper [Theorem 1, https://eprint.iacr.org/2022/1696.pdf], we roughly have the following:

Adv^{IND-CCA} <= 2 q_{RO} \sqrt{Adv^{IND-CPA}} + other terms,

which led to the above numbers (i.e., 2^-166, 2^-136). However, following the recent injectivity analysis of Kyber by Ding et. al. [https://link.springer.com/chapter/10.1007/978-3-031-22301-3_17], our proof strategy allows for much tighter bounds w.r.t. Kyber. For instance, the QROM analysis of the (implicitly rejecting) FO transforms by Bindel et. al. [https://eprint.iacr.org/2019/590.pdf] leads to the following bounds for Kyber:

Adv^{IND-CCA} <= 2 \sqrt{q_{RO} Adv^{IND-CPA}} + other terms,

which essentially brings down the above IND-CPA advantage numbers to 2^-82 and 2^-67 respectively.

Zooming out, we are not suggesting that Kyber needs to update its parameters following our IND-CCA security proof in the QROM. Security reductions in the QROM are notorious for their non-tightness, and we believe further work needs to be done to tame this non-tightness before scheme designers can set parameters based on the corresponding security proof (rather than the best known attacks on the underlying hardness assumption).

Our work instead showcases that the state-of-the-art concrete bounds for the (implicitly rejecting) FO transforms in the QROM also apply to Kyber.

Best,

Varun

On Wednesday, January 18, 2023 at 5:22:29 PM UTC+5:30 D. J. Bernstein wrote:

> Varun Maram writes:
> > Along the way, we also provide a proof of IND-CCA security for Kyber
> > in the QROM with concrete bounds
>
> To clarify, am I correctly understanding that
>
> * the proof assumes IND-CPA security for the underlying PKE, and
>
> * the concrete bounds for Kyber-1024 (or Kyber-512) apply only if the
> attacker is limited to 2^82 (or 2^67) hash calls and doesn't have
> an IND-CPA attack with success chance above 2^-166 (or 2^-136)?
>
> Is there literature claiming that such low-probability IND-CPA attacks
> are infeasible against Kyber, and quantifying the claimed security level
> for comparison to NIST's security requirements?
>
> I realize that integrating depth limits into the proof should change
> the numbers somewhat, but in any case applying the proof strategy seems
> to require an analysis of low-probability attacks, and I don't see why
> those should be assumed to cost as much as high-probability attacks. My
> recent paper https://cr.yp.to/papers.html#lprrr shows one way to exploit
> this difference for asymptotically faster attacks assuming standard
> heuristics. A concrete analysis is harder than an asymptotic analysis,
> but I'd presume that Kyber loses security from the same effect.
>
> ---D. J. Bernstein

To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/0078b5e3-168e-4252-abd3-fb80e4b5c5den%40list.nist.gov.

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Kyber decisions, part 2: FO transform
Date: Friday, January 20, 2023 10:31:03 AM ET
Attachments: smime.p7m

Varun Maram writes:

> which essentially brings down the above IND-CPA advantage numbers to
> 2^-82 and 2^-67 respectively.

Thanks for the calculations. This sets concrete targets for the missing analysis of low-probability IND-CPA attacks against Kyber. I'd also suggest making a table of how the targets change if the user isn't willing to accept IND-CCA2 attacks with probability above, say, 2^-10.

> Security reductions in the QROM are notorious for their non-tightness

There's an important exception here: implicit-rejection KEMs built from correct _deterministic_ PKEs. For this case, the 2019 BHHHP theorem says that the QROM IND-CCA2 advantage is at most

   2 sqrt(OW-CPA success probability against the PKE)
   + 2 (#hash queries) / sqrt(#implicit-rejection keys).

This starts saying something as soon as the OW-CPA success probability drops below (about) 2^-2, and it says that the QROM IND-CCA2 advantage drops below 2^-10 when the OW-CPA success probability drops below 2^-22.

This special class of KEMs doesn't include Kyber, but it does include Classic McEliece and the two NTRU variants that are now most widely deployed (ntruhrss701 and sntrup761). There are also clear differences in the status of the cryptanalysis of the underlying assumptions:

   * Classic McEliece: The quantitative analysis of how OW-CPA attack
     costs drop with probability is easy, and is included in the
     documentation.

* ntruhrss701, sntrup761: The literature seems to be missing an
  analysis of concrete OW-CPA attack costs for low success
  probabilities, such as 2^-22. This is concerning. (Even without
  quantum attacks, where's the analysis of, e.g., probability 2^-10?)

* Kyber: The literature seems to be missing an analysis of concrete
  IND-CPA attacks for very low success probabilities, such as 2^-82.
  This is even more concerning than the NTRU situation, in part
  because IND-CPA is more complicated than OW-CPA and in part because
  2^82 is far beyond 2^20.

Alarm bells should be going off when there's a mismatch between the
assumptions made in a claim of provable security and the assumptions
that cryptanalysts have been trying to break.

> we believe further work needs to be done to tame this non-tightness
> before scheme designers can set parameters based on the corresponding
> security proof (rather than the best known attacks on the underlying
> hardness assumption).

When proof looseness allows a low-probability attack against underlying
hardness assumptions to turn into a high-probability attack against the
cryptosystem, it's dangerous to ignore this gap. (Same for time gaps.)
See https://www.youtube.com/watch?v=l56ORg5xXkk for a nice introduction
to the "nightmare scenario" for these proofs.

Accounting for proof looseness in cryptosystem selection would have
avoided the collapse of security claims for MQDSS and FrodoKEM.

——D. J. Bernstein

--

From: Mike Hamburg <mike@shiftleft.org> via pqc-forum@list.nist.gov
To: D. J. Bernstein <djb@cr.yp.to>
CC: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Kyber decisions, part 2: FO transform
Date: Sunday, January 22, 2023 03:32:33 PM ET

> On Jan 20, 2023, at 4:30 PM, D. J. Bernstein <djb@cr.yp.to> wrote:
>
> * Kyber: The literature seems to be missing an analysis of concrete IND-CPA attacks for very low success probabilities, such as 2^-82. This is even more concerning than the NTRU situation, in part because IND-CPA is more complicated than OW-CPA and in part because 2^82 is far beyond 2^20.
>
> Alarm bells should be going off when there's a mismatch between the assumptions made in a claim of provable security and the assumptions that cryptanalysts have been trying to break.

Hi Dan, hi all,

It may be worth mentioning that BHHHP's bound for rPKE uses

IND-CPA because that's a standard assumption. However, the

attacker's goal in that theorem is probably better understood as

OW-CPA with a semi-classical correctness oracle. Consider, that

is, the following "OW-CPA-scChk" game against an rPKE:

def OW-CPA-scChk(A):

(sk, pk) = keygen()

m = random message

r = random coins

c = rPKE(pk, m, r)

def Chk(mm): # semiclassical checking oracle

find = (mm == m)

measure find

if find: abort

A(Chk, pk, c)

The adversary's goal is to cause the semiclassical Chk oracle to abort by passing it m.

The adversary's success probability at OW-CPA on derandomized PKE is at most (d+2) times higher than the OW-CPA-scChk advantage, unless I've screwed up. I'm rusty, so feel free to check me on this; it should be straightforward from AHU'18 Theorem 1, aka semiclassical one-way to hiding.

Proof sketch:

==========

* Reformulate the OW-CPA game so that the coins r are random instead of being G(m), but G(m) is reprogrammed to return r. This changes nothing; it's just a formality to move the setup of the derandomized OW-CPA one step closer to rPKE OW-CPA-scChk.

* At the end of the OW-CPA game, add a step that calls G on the adversary's output before returning. This changes nothing except to increase the G-queries from q to q+1 and the depth from d to d+1.

* Puncture the oracle by replacing G(x) with { Chk(x); G(x) }. The
punctured game is rather interesting:

** The attacker can't win OW-CPA in the punctured game, because
that last added G-call would cause an abort.

** In the punctured game, the reprogramming G(m) = r is never used
and can be dropped, because you've first checked that x != m.
Therefore, in the punctured game, the coins are only used in the
challenge ciphertext c, and the message is only used to create c
and the oracle Chk.

** At this point, the random oracle G also isn't used by the challenger.
The adversary can still call it to get random numbers though.

* The adversary's advantage in the real OW-CPA game is therefore
at most (d+2) Pfind by semiclassical O2H (the "difference of
square roots" version, with bound (3)).

* Pfind is the OW-CPA-scChk advantage of an attacker running in
very nearly the same resources as A.

==========

Since OW-CPA-scChk is a nonstandard assumption, BHHHP
instead reduces to IND-CPA by puncturing on two messages.
You should also be able to convert to vanilla OW-CPA against
the rPKE, but at the cost of another leading 4(q+1) or so from
AHU'19 Theorem 2 (or likely by invoking quantum O2H instead of
the semiclassical version, but either way the leading term should

be about 4dq).

The leading term is still a potential concern, but it's salient that the OW-CPA-scChk goal is is not inherently easier than deterministic OW-CPA; it may instead be harder. That is, in deterministic OW-CPA, the attacker can check whether a given guess is correct in superposition, but here the attacker only has a semiclassical oracle. Of course, I expect that the true difficulty will mostly depend on the PKE itself.

The problem of breaking one-way-ness in Kyber or other LPR-like schemes with a semiclassical checking oracle hasn't been studied to my knowledge, but it is qualitatively an OW problem and not an IND problem. So at least that part of the concern is mitigated somewhat.

Regards,

— Mike

[AHU'19] Ambainis, Hamburg, Unruh. Quantum security proofs using semi-classical oracles. https://eprint.iacr.org/2018/904.pdf

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Kyber decisions, part 2: FO transform
Date: Tuesday, January 24, 2023 12:11:48 PM ET
Attachments: smime.p7m

Mike Hamburg writes:

> it is qualitatively an OW problem and not an IND problem. So at least
> that part of the concern is mitigated somewhat.

Can you please explain how it's mitigated?

The stated reason for concern was that "IND-CPA is more complicated than OW-CPA". Obviously this "scChk" hypothesis is also more complicated than OW-CPA. I don't see how the switch of hypothesis addresses the concern.

I agree that distinguishers are a complication avoided by the scChk definition, but the scChk definition has its own quantum complications, so it's not as if there's some clear attraction for cryptanalysts.

Meanwhile the broader reason stated for "alarm bells" was the "mismatch between the assumptions made in a claim of provable security and the assumptions that cryptanalysts have been trying to break". Switching to scChk exacerbates this concern today, even if scChk turns out to attract cryptanalysis in the long term.

——D. J. Bernstein