From: Moody, Dustin (Fed) <dustin.moody@nist.gov> via pqc-forum <pqc-forum@list.nist.gov>
To: pqc-forum <pqc-forum@list.nist.gov>
Subject: [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, November 30, 2022 07:26:01 AM ET

Dear all,

NIST has been thinking a lot about which parameter sets to include as part of the standards for the selected algorithms. We wanted to share our current view.

Dilithium

We are planning to include parameter sets corresponding to security categories 2, 3, and 5.We are not planning to include the AES variant.

Falcon

We are planning to include parameter sets corresponding to security categories 1 and 5.

SPHINCS+

We plan to include parameter sets for security categories 1, 3, and 5.We plan to include the simple (and NOT the robust) version.We also plan to include both the fast and small versions.Allowed hash functions will be SHAKE and SHA-2(SHA-256 for category 1 and a mix of SHA-512 and SHA-256 for categories 3 and 5).

Kyber

We are planning on including parameter sets for categories 1, 3, and 5, though we would highly recommend the category 3 parameter set as the default option.We are not planning on including the 90s variant.

The decision for the category 1 parameter set has been more difficult.There have been extensive discussions if and in what metrics this parameter set achieves the same security as AES-128. It is clear that in the gate-count metric it is a very close call and that in this metric the pre-quantum security of Kyber-512 may be a few bits below the one of AES-128. However, the best known attacks against Kyber-512 require huge amounts of memory and the real attack cost will need to take the cost of (access to) memory into account.This cost is not easy to calculate, as it depends on the memory access patterns of the lattice algorithms used for cryptanalysis, as well as the physical properties of the memory hardware. Nonetheless, barring major improvements in cryptanalysis, it seems unlikely that the cost of memory access will ever become small enough to cause Kyber-512 to fall below category 1 security, in realistic models of security that take these costs into account. We acknowledge there can be different views on our current view to include Kyber-512.

As a point of clarification: in this email, we refer to parameter sets based on the claimed security strength category where those parameter sets are most recently specified, irrespective of whether those parameter sets actually meet their claimed security level. That said, our current assessment is that, when realistic memory access costs of known attacks are taken into account, all the parameter sets we plan to standardize do, in fact, meet their claimed security strength categories.

NIST PQC team

From: hi@arnepadmos.com
To: Moody, Dustin (Fed) <dustin.moody@nist.gov>
CC: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, November 30, 2022 11:13:32 AM ET

Dear NIST PQC team,

Thank you for this update. Quoting from page 18 of the third round report, which covers potential adoption challenges posed by third-party patents: 'NIST expects to execute the various agreements prior to publishing the standard. If the agreements are not executed by the end of 2022, NIST may consider selecting NTRU instead of KYBER.' Are there any updates on this front?

Regards,
Arne

On 2022-11-30 13:25, 'Moody, Dustin (Fed)' via pqc-forum wrote:
> Dear all,
>
>
> NIST has been thinking a lot about which parameter sets to include as
> part of the standards for the selected algorithms. We wanted to share
> our current view.
>
>
> Dilithium
>
> We are planning to include parameter sets corresponding to security
> categories 2, 3, and 5. We are not planning to include the AES
> variant.
>
>
>
> Falcon
>

> We are planning to include parameter sets corresponding to security
> categories 1 and 5.
>
>
>
> SPHINCS+
>
> We plan to include parameter sets for security categories 1, 3, and 5.
> We plan to include the simple (and NOT the robust) version. We also
> plan to include both the fast and small versions. Allowed hash
> functions will be SHAKE and SHA-2 (SHA-256 for category 1 and a mix of
> SHA-512 and SHA-256 for categories 3 and 5).
>
>
>
> Kyber
>
> We are planning on including parameter sets for categories 1, 3, and
> 5, though we would highly recommend the category 3 parameter set as
> the default option. We are not planning on including the 90s variant.
>
>
>
> The decision for the category 1 parameter set has been more difficult.
> There have been extensive discussions if and in what metrics this
> parameter set achieves the same security as AES-128. It is clear that
> in the gate-count metric it is a very close call and that in this
> metric the pre-quantum security of Kyber-512 may be a few bits below
> the one of AES-128. However, the best known attacks against Kyber-512
> require huge amounts of memory and the real attack cost will need to
> take the cost of (access to) memory into account. This cost is not
> easy to calculate, as it depends on the memory access patterns of the
> lattice algorithms used for cryptanalysis, as well as the physical
> properties of the memory hardware. Nonetheless, barring major
> improvements in cryptanalysis, it seems unlikely that the cost of
> memory access will ever become small enough to cause Kyber-512 to fall
> below category 1 security, in realistic models of security that take

> these costs into account. We acknowledge there can be different views
> on our current view to include Kyber-512.
>
>
>
>
> As a point of clarification: in this email, we refer to parameter sets
> based on the claimed security strength category where those parameter
> sets are most recently specified, irrespective of whether those
> parameter sets actually meet their claimed security level. That said,
> our current assessment is that, when realistic memory access costs of
> known attacks are taken into account, all the parameter sets we plan
> to standardize do, in fact, meet their claimed security strength
> categories.
>
>
> NIST PQC team

From: Moody, Dustin (Fed) <dustin.moody@nist.gov> via pqc-forum <pqc-forum@list.nist.gov>
To: hi@arnepadmos.com
CC: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, November 30, 2022 11:19:35 AM ET

Arne,

Yes, we did share some updates in this direction yesterday at our workshop.

The license agreements mentioned in NISTIR 8413 have been signed by all parties. NIST appreciates the efforts of those who helped obtain this outcome and the cooperation of third parties. (CNRS, the University of Limoges, the laboratory XLIM, and Jintai Ding)

The relevant text of the license can be found at:

https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/selected-algos-2022/nist-pqc-license-summary-and-excerpts.pdf

The license allows for royalty-free use (from the third parties listed above) of implementations which follow the NIST standard. [Disclaimer - I'm not a lawyer, so see the exact text from the link posted above for precise details.]

NIST is not considering NTRU for standardization.

Thanks,

Dustin Moody

From: hi@arnepadmos.com <hi@arnepadmos.com>
Sent: Wednesday, November 30, 2022 11:13 AM
To: Moody, Dustin (Fed) <dustin.moody@nist.gov>
Cc: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms

Dear NIST PQC team,

Thank you for this update. Quoting from page 18 of the third round report, which covers potential adoption challenges posed by third-party patents: 'NIST expects to execute the various agreements prior to publishing the standard. If the agreements are not executed by the end of 2022, NIST may consider selecting NTRU instead of KYBER.' Are there any updates on this front?

Regards,
Arne

On 2022-11-30 13:25, 'Moody, Dustin (Fed)' via pqc-forum wrote:
> Dear all,
>
>
> NIST has been thinking a lot about which parameter sets to include as
> part of the standards for the selected algorithms. We wanted to share
> our current view.
>
>
> Dilithium
>
> We are planning to include parameter sets corresponding to security
> categories 2, 3, and 5. We are not planning to include the AES
> variant.
>
>
>
> Falcon
>
> We are planning to include parameter sets corresponding to security
> categories 1 and 5.
>
>
>
> SPHINCS+
>
> We plan to include parameter sets for security categories 1, 3, and 5.
> We plan to include the simple (and NOT the robust) version. We also
> plan to include both the fast and small versions. Allowed hash
> functions will be SHAKE and SHA-2 (SHA-256 for category 1 and a mix of
> SHA-512 and SHA-256 for categories 3 and 5).

>
>
>
> Kyber
>
> We are planning on including parameter sets for categories 1, 3, and
> 5, though we would highly recommend the category 3 parameter set as
> the default option. We are not planning on including the 90s variant.
>
>
>
> The decision for the category 1 parameter set has been more difficult.
> There have been extensive discussions if and in what metrics this
> parameter set achieves the same security as AES-128. It is clear that
> in the gate-count metric it is a very close call and that in this
> metric the pre-quantum security of Kyber-512 may be a few bits below
> the one of AES-128. However, the best known attacks against Kyber-512
> require huge amounts of memory and the real attack cost will need to
> take the cost of (access to) memory into account. This cost is not
> easy to calculate, as it depends on the memory access patterns of the
> lattice algorithms used for cryptanalysis, as well as the physical
> properties of the memory hardware. Nonetheless, barring major
> improvements in cryptanalysis, it seems unlikely that the cost of
> memory access will ever become small enough to cause Kyber-512 to fall
> below category 1 security, in realistic models of security that take
> these costs into account. We acknowledge there can be different views
> on our current view to include Kyber-512.
>
>
>
>
> As a point of clarification: in this email, we refer to parameter sets
> based on the claimed security strength category where those parameter
> sets are most recently specified, irrespective of whether those
> parameter sets actually meet their claimed security level. That said,
> our current assessment is that, when realistic memory access costs of
> known attacks are taken into account, all the parameter sets we plan
> to standardize do, in fact, meet their claimed security strength
> categories.
>

>
> NIST PQC team

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: [pqc-forum] Re: patents
Date: Friday, December 02, 2022 07:19:12 AM ET
Attachments: smime.p7m

Regarding the patent licenses, I asked the following clarification questions at the conference: "It's great to see the progress in opening things up for free worldwide use. I have two questions: (1) Suppose someone deploys the current version of Kyber-768 (the round-3 version), and then in 2024 NIST standardizes a tweaked version of Kyber-768 (maybe different hash functions or prefix hashing). Am I correctly understanding the first license to say that the license is only for the tweaked version, not today's version, so the patent holder can sue for infringement regarding the deployment? (2) Are these two agreements the end of NIST's patent negotiations regarding Kyber---i.e., NIST has no negotiations underway regarding any of the other patent families listed in https://ntruprime.cr.yp.to/faq.html?"

The NIST representative said he would have to check with NIST's legal team before answering. I'm looking forward to seeing the answers here.

Various people have also pointed out a further clarification question that's important to answer: (3) _When_ exactly do the licenses permit Kyber deployment under those two patent families? Consider, for example, the following scenario:

* NIST announces at some point in 2023 something like "We're putting an end to the Kyber modifications, and the final Kyber to be standardized as NIST MLWE-KEM is as follows." This handles #1.

* Other patent threats also turn out to have been resolved by then (through buyouts, or through sufficient analysis for the public to be sure that the patents don't apply). This handles #2.

* A company then asks whether it can safely deploy that version of Kyber in 2023, or whether it has to wait until the standard is

issued in 2024.

The text posted by NIST says "any time on or after the EFFECTIVE DATE" but the definition of "EFFECTIVE DATE" seems to have been omitted. The text is labeled as "relevant language (with modifications for readability) from the licenses" but doesn't seem to have _all_ relevant language. Even if the "EFFECTIVE DATE" is in the past, isn't the company infringing the patent in 2023 since at that point there's no "standard prescribed by NIST"? It's not clear to me from the existing text how the issuance of a standard could retroactively remove earlier infringement.

As a procedural matter, back in July, Scott Fluhrer wrote "until we get the text of the licenses (both the one signed with CNRS and the one to be signed with Algo Consulting), Cisco cannot use Kyber". It's hard to imagine a corporate lawyer viewing edited excerpts as an adequate substitute for the full, unedited license text; hidden contract provisions can override other provisions. There also seems to be some confusion regarding the overall status of the license text (e.g., the NIST representative writing "I'm pretty sure what is posted is verbatim text" while the NIST text says "modifications for readability"), and lawyers tend to go on high alert when they see such inconsistencies.

---D. J. Bernstein

P.S. The first time I sent this message, Google bounced it, saying "The group pqc-forum@list.nist.gov has exceeded its quota for total number of external recipients." So I'm trying again now. Meanwhile it seems that a message from hi@arnepadmos.com wasn't delivered to the list.

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Friday, December 02, 2022 08:54:33 AM ET
Attachments: smime.p7m

'Moody, Dustin (Fed)' via pqc-forum writes:

> Nonetheless, barring major improvements in cryptanalysis, it seems
> unlikely that the cost of memory access will ever become small enough
> to cause Kyber-512 to fall below category 1 security, in realistic
> models of security that take these costs into account.

Please clarify. My current guess is that NIST is stating the following:

(A) In reality, because of the costs of memory _relative to computation_, all published Kyber-512 attack algorithms are much more expensive than single-target AES-128 key search.

(B) It seems unlikely for the real costs of memory relative to computation to improve much.

(C) Ergo, in reality, breaking Kyber-512 more cheaply than single-target AES-128 key search would require much better attack algorithms than what has been published.

My current guess is that NIST is _not_ making the following statement, a statement very different from B:

(D) It seems unlikely that there are much better attack algorithms against Kyber-512.

But perhaps the scope of "unlikely" was meant not to refer not merely to the real costs of memory but also to the "barring" hypothesis. Also, perhaps I'm misunderstanding what "small enough" was meant to indicate. Clarification would be useful for cryptanalysts and for decisionmakers interested in long-term security.

---D. J. Bernstein

--

From: Derek Atkins <datkins@veridify.com> via pqc-forum@list.nist.gov
To: djb@cr.yp.to, pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Friday, December 02, 2022 09:24:50 AM ET

Dan,

I don't think any clarification is required. I think it is quite obvious that NIST is claiming A, B, and C, and is not claiming D, just as you "guess"ed.

-derek

On Fri, 2022-12-02 at 14:54 +0100, D. J. Bernstein wrote:

> 'Moody, Dustin (Fed)' via pqc-forum writes:
>
> > Nonetheless, barring major improvements in cryptanalysis, it seems
> >
> > unlikely that the cost of memory access will ever become small enough
> >
> > to cause Kyber-512 to fall below category 1 security, in realistic
> >
> > models of security that take these costs into account.
>
> Please clarify. My current guess is that NIST is stating the following:
>
> (A) In reality, because of the costs of memory _relative to
>
> computation_, all published Kyber-512 attack algorithms are much
>
> more expensive than single-target AES-128 key search.
>
> (B) It seems unlikely for the real costs of memory relative to
>
> computation to improve much.
>
> (C) Ergo, in reality, breaking Kyber-512 more cheaply than
>
> single-target AES-128 key search would require much better attack
>
> algorithms than what has been published.
>
> My current guess is that NIST is _not_ making the following statement, a

> statement very different from B:
>
> (D) It seems unlikely that there are much better attack algorithms
>
> against Kyber-512.
>
> But perhaps the scope of "unlikely" was meant not to refer not merely to
>
> the real costs of memory but also to the "barring" hypothesis. Also,
>
> perhaps I'm misunderstanding what "small enough" was meant to indicate.
>
> Clarification would be useful for cryptanalysts and for decisionmakers
>
> interested in long-term security.
>
> ---D. J. Bernstein

--

Derek Atkins
Chief Technology Officer
Veridify Security - *Securing the Internet of Things®*

Office: 203.227.3151 x1343
Direct: 617.623.3745
Mobile: 617.290.5355
Email: DAtkins@Veridify.com

This email message may contain confidential, proprietary and / or legally privileged information and intended only for the use of the intended recipient(s) and others specifically authorized. Any disclosure, dissemination, copying, distribution or use of the information contained in this email message, including any attachments, to or by anyone other than the intended recipient is strictly prohibited. If you received this in error, please immediately advise the sender by reply email or at the telephone number above, and then delete, shred, or otherwise dispose of this message.

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Friday, December 02, 2022 10:39:16 AM ET
Attachments: smime.p7m

Derek Atkins writes:

> I don't think any clarification is required. I think it is quite
> obvious that NIST is claiming A, B, and C, and is not claiming D, just
> as you "guess"ed.

Hmmm. Taking the original words "small enough" literally wouldn't match what A says; and readers who aren't sticking to literal interpretations could easily understand the "barring ... unlikely" part to indicate D. I think A+B+C-not-D is a reasonable interpretation, but I'm not at all sure that this is what NIST meant. NIST should clarify.

---D. J. Bernstein

From: John Mattsson <john.mattsson@ericsson.com> via pqc-forum <pqc-forum@list.nist.gov>
To: Moody, Dustin (Fed) <dustin.moody@nist.gov>, hi@arnepadmos.com
CC: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Friday, December 02, 2022 12:33:59 PM ET

I think this seems like a good plan. I assume the suggested changes to the algorithms like the excellent TurboSHAKE idea will be discussed in a separate thread.

>Kyber-512 may be a few bits below the one of AES-128.

I don't think this is a problem in practice. The same is true for Curve25519. I think "realistic models of security" are the only one NIST should consider when making decisions. I think NIST should standardize Kyber-512 andlabel Kyber-512 as Level I. The public keys and encapsulations in Kyber-768 is a whopping 400 bytes larger which will decrease performance.

BTW, when will we get any news on FIPS 186-5 and SP 800-186? These are very important specifications. RSA and ECC will continue being used for decades, both standalone and as part of hybrid solutions. I would like to see them published yesterday. It seems obvious that the specifications has met some opposition in the publication process. Is NIST planning to give any updates on the situation?

Cheers,

John

From: 'Moody, Dustin (Fed)' via pqc-forum <pqc-forum@list.nist.gov>
Date: Wednesday, 30 November 2022 at 17:21
To: hi@arnepadmos.com <hi@arnepadmos.com>
Cc: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms

Arne,

Yes, we did share some updates in this direction yesterday at our workshop.

The license agreements mentioned in NISTIR 8413 have been signed by all parties. NIST appreciates the efforts of those who helped obtain this outcome and the cooperation of third parties. (CNRS, the University of Limoges, the laboratory XLIM, and Jintai Ding)

The relevant text of the license can be found at:

https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/selected-algos-2022/nist-pqc-license-summary-and-excerpts.pdf

The license allows for royalty-free use (from the third parties listed above) of implementations which follow the NIST standard. [Disclaimer - I'm not a lawyer, so see the exact text from the link posted above for precise details.]

NIST is not considering NTRU for standardization.

Thanks,

Dustin Moody

From: hi@arnepadmos.com <hi@arnepadmos.com>
Sent: Wednesday, November 30, 2022 11:13 AM
To: Moody, Dustin (Fed) <dustin.moody@nist.gov>
Cc: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms

Dear NIST PQC team,

Thank you for this update. Quoting from page 18 of the third round report, which covers potential adoption challenges posed by third-party patents: 'NIST expects to execute the various agreements prior to publishing the standard. If the agreements are not executed by the end of 2022, NIST may consider selecting NTRU instead of KYBER.' Are there any updates on this front?

Regards,
Arne

On 2022-11-30 13:25, 'Moody, Dustin (Fed)' via pqc-forum wrote:
> Dear all,
>
>
> NIST has been thinking a lot about which parameter sets to include as
> part of the standards for the selected algorithms. We wanted to share
> our current view.

>
>
> Dilithium
>
> We are planning to include parameter sets corresponding to security
> categories 2, 3, and 5. We are not planning to include the AES
> variant.
>
>
>
> Falcon
>
> We are planning to include parameter sets corresponding to security
> categories 1 and 5.
>
>
>
> SPHINCS+
>
> We plan to include parameter sets for security categories 1, 3, and 5.
> We plan to include the simple (and NOT the robust) version. We also
> plan to include both the fast and small versions. Allowed hash
> functions will be SHAKE and SHA-2 (SHA-256 for category 1 and a mix of
> SHA-512 and SHA-256 for categories 3 and 5).
>
>
>
> Kyber
>
> We are planning on including parameter sets for categories 1, 3, and
> 5, though we would highly recommend the category 3 parameter set as
> the default option. We are not planning on including the 90s variant.
>
>
>
> The decision for the category 1 parameter set has been more difficult.
> There have been extensive discussions if and in what metrics this

> parameter set achieves the same security as AES-128. It is clear that
> in the gate-count metric it is a very close call and that in this
> metric the pre-quantum security of Kyber-512 may be a few bits below
> the one of AES-128. However, the best known attacks against Kyber-512
> require huge amounts of memory and the real attack cost will need to
> take the cost of (access to) memory into account. This cost is not
> easy to calculate, as it depends on the memory access patterns of the
> lattice algorithms used for cryptanalysis, as well as the physical
> properties of the memory hardware. Nonetheless, barring major
> improvements in cryptanalysis, it seems unlikely that the cost of
> memory access will ever become small enough to cause Kyber-512 to fall
> below category 1 security, in realistic models of security that take
> these costs into account. We acknowledge there can be different views
> on our current view to include Kyber-512.
>
>
>
>
> As a point of clarification: in this email, we refer to parameter sets
> based on the claimed security strength category where those parameter
> sets are most recently specified, irrespective of whether those
> parameter sets actually meet their claimed security level. That said,
> our current assessment is that, when realistic memory access costs of
> known attacks are taken into account, all the parameter sets we plan
> to standardize do, in fact, meet their claimed security strength
> categories.
>
>
> NIST PQC team

--

You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/SA1PR09MB86694461EA7AA74FC0A5A4A5E5159%40SA1PR09MB8669.namprd09.prod.outlook.com.

From: dustin...@nist.gov <dustin.moody@nist.gov> via pqc-forum <pqc-forum@list.nist.gov>
To: pqc-forum <pqc-forum@list.nist.gov>
CC: D. J. Bernstein <djb@cr.yp.to>, pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Friday, December 02, 2022 02:03:38 PM ET

Hi Dan,

What we meant was: There are various possible improvements in cryptanalysis, including: (i) optimization of lattice algorithms to improve the locality of memory access, (ii) building memory hardware that reduces the cost of non-local memory access, and (iii) other (more fundamental) improvements in the attacks. It seems unlikely that (i) and (ii) alone will be sufficient to cause Kyber-512 to fall below category 1 security, in realistic models of security that take these costs into account.

Dustin

On Friday, December 2, 2022 at 10:39:13 AM UTC-5 D. J. Bernstein wrote:

> Derek Atkins writes:
> \> I don't think any clarification is required. I think it is quite
> \> obvious that NIST is claiming A, B, and C, and is not claiming D, just
> \> as you "guess"ed.
>
> Hmmm. Taking the original words "small enough" literally wouldn't match
> what A says; and readers who aren't sticking to literal interpretations
> could easily understand the "barring ... unlikely" part to indicate D. I
> think A+B+C-not-D is a reasonable interpretation, but I'm not at all
> sure that this is what NIST meant. NIST should clarify.
>
> ---D. J. Bernstein

To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/37c5425e-ef1d-4015-ada7-ae6c4673b31an%40list.nist.gov.

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Saturday, December 03, 2022 11:12:41 AM ET
Attachments: smime.p7m

'dustin...@nist.gov' via pqc-forum writes:

> What we meant was: There are various possible improvements in
> cryptanalysis, including: (i) optimization of lattice algorithms to improve
> the locality of memory access, (ii) building memory hardware that reduces
> the cost of non-local memory access, and (iii) other (more fundamental)
> improvements in the attacks. It seems unlikely that (i) and (ii) alone will
> be sufficient to cause Kyber-512 to fall below category 1 security, in
> realistic models of security that take these costs into account.

Thanks for the clarification.

In the interests of transparency and falsifiability, can NIST please explain how it arrived at the above likelihood assessment? The pieces sound like

(1) thinking that known Kyber-512 attacks haven't dropped far below single-target AES-128 key search (2^143) in gate counts,

(2) thinking that the costs of memory gain many more bits of security against those Kyber-512 attacks,

(3) thinking that improvements in algorithmic memory locality are unlikely to reduce Kyber-512's security level by much, and

(4) thinking that improvements in physical memory hardware (compared to computation) are unlikely to reduce Kyber-512's security level by much,

but I don't see any documentation of how NIST arrived at any of these individual conclusions, never mind evaluating the likelihood of error in the overall conclusion.

Of course, I checked the latest Kyber document from the Kyber team:

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

This document addresses #1 but is obviously outdated, estimating 151 where typical estimates today are in the 130s. Which of the newer attacks and analyses is NIST taking into account? Is NIST's evaluation of #1 accounting for "known unknowns", and, if so, how?

As for the costs of memory, the 2021 Kyber document makes a brief, unquantified claim that 135 wouldn't be "catastrophic, in particular given the massive memory requirements that are ignored in the gate-count metric". Is this somehow the basis of NIST arriving at #2 or #3 or #4?

---D. J. Bernstein

From: peter...@gmail.com <peter.pessl@gmail.com> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
CC: dustin...@nist.gov <dustin.moody@nist.gov>
Subject: [pqc-forum] Re: Parameter selection for the selected algorithms
Date: Monday, December 05, 2022 11:33:30 AM ET

Hi Dustin,
dustin...@nist.gov schrieb am Mittwoch, 30. November 2022 um 13:25:55 UTC+1:

> Dilithium
> We are planning to include parameter sets corresponding to security categories 2, 3, and 5.We are not planning to include the AES variant.

there is one additional degree of freedom for Dilithium, namely deterministic vs. randomized signing (described in Section 3.2 of the current Dilithium specification). Since I didn't see it mentioned by NIST, I'd like to ask: does NIST intend to standardize both modes? If yes, will there be "preferred mode" or are implementers free to choose for all applications?

BR Peter

From: Perlner, Ray A. (Fed) <ray.perlner@nist.gov> via pqc-forum <pqc-forum@list.nist.gov>
To: D. J. Bernstein <djb@cr.yp.to>, pqc-forum <pqc-forum@list.nist.gov>
Subject: RE: [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, December 07, 2022 05:40:09 PM ET

Hi Dan,

We can elaborate a little bit further on our reasoning leading to our current assessment that Kyber512 likely meets NIST category I (similar considerations apply to the other parameter sets we plan to standardize for lattice-based schemes.) That said, beyond this message, we don’t think further elaboration of our current position will be helpful. While we did consult among ourselves and with the Kyber team, it’s basically just our considered opinion based on the same publicly available information everyone else has access to. The point of this thread is to seek a broader range of perspectives on whether our current plan to standardize Kyber512 is a good one, and a long back and forth between us and a single researcher does not serve that purpose.

Here's how we see the situation:

In April this year, “Report on the Security of LWE” was published by MATZOV (https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fzenodo.org%2Frecord%2F6412487%23.Y4-V53bMKUk&amp;data=05%7C01%7Cyi-kai.liu%40nist.gov%7C5698dfa237584e404afe08dad8a3fd48%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638060496089215178%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=dGegz0B9hGFzObtf7IHkz4vm64UGUrzGGeqLwUOZJrk%3D&amp;reserved=0), describing an attack, assessed in the RAM model to bring some parameter sets, including Kyber512, slightly below their claimed security strength categories. In particular, the report estimates the cost of attacking Kyber512 using a classical lattice attack to be 2^137 bit operations, which is less than the approximately 2^143 bit operations required to classically attack AES-128. However, like previous lattice attacks, the MATZOV attack is based on sieving techniques, which require a large amount of (apparently unstructured) access to a very large memory. The RAM model ignores the cost of this memory access, and while the science of comparing the cost of memory access to other costs involved in a large cryptanalytic attack is not as mature as we would like, it seems overwhelmingly likely that, in any realistic accounting of memory access costs, these will significantly exceed the costs that are assessed by the RAM model for lattice sieving.

The largest practical implementation of sieving techniques we know of, described in detail in “Advanced Lattice Sieving on GPUs, with Tensor Cores” by Ducas, Stevens, and van Woerden (https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Feprint.iacr.org%2F2021%2F141&amp;data=05%7C01%7Cyi-kai.liu%40nist.gov%7C5698dfa237584e404afe08dad8a3fd48%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638060496089215178%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=rQU21hLbLZN%2B5ys3NJYtHRMqy9Mhtm2tPOwzrI0AXmI%3D&amp;reserved=0), was forced by memory access limitations, to adopt settings for bucket size, that would be suboptimal according to the RAM model. It should be noted that, increasing the scale of the instances being attacked to near cryptographic scale would probably require extensive hardware optimization, e.g. by using special purpose ASICs, and these techniques, being generally acknowledged to be less effective against memory-intensive tasks, would likely make memory access even more of a bottleneck.

Additionally, While the Kyber, Dilithium, and Falcon teams did not give a quantitative assessment of the practical cost of memory access during sieving against cryptographic parameters, assessments by the NTRU and NTRUprime teams gave estimates that would suggest the cost of sieving against category 1 parameters, in models that account for the cost of memory access, is something like 20 to 40 bits of security more than would be suggested by the RAM model. (For NTRU’s estimates see section 6.3 of the round 3 specification document available at https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fntru.org%2Findex.shtml&amp;data=05%7C01%7Cyi-kai.liu%40nist.gov%7C5698dfa237584e404afe08dad8a3fd48%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638060496089215178%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=6CEF%2BvETZu19V7mspw5ellp8mCvd7DmdDB8ImjygESY%3D&amp;reserved=0 . For NTRUprime’s estimates see section 6.11 of https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fntruprime.cr.yp.to%2Fnist%2Fntruprime-20201007.pdf&amp;data=05%7C01%7Cyi-kai.liu%40nist.gov%7C5698dfa237584e404afe08dad8a3fd48%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638060496089215178%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=JhDAzOUbSPJcaudnz2wR6Vd6XE9OQgtVuB%2FzLyICOb0%3D&amp;reserved=0 . The Kyber spec (available at https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fpq-crystals.org%2Fkyber%2Fdata%2Fkyber-specification-round3-20210804.pdf&amp;data=05%7C01%7Cyi-kai.liu%40nist.gov%7C5698dfa237584e404afe08dad8a3fd48%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638060496089215178%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=suRdOS%2Btt3PM5DO%2BQiCiaAIfusZaoTqvEl5iWb%2BqUNU%3D&amp;reserved=0) discusses, but does not quantify, memory access costs in section 5.3 (Q6))

Taking Matzov's estimates of the attack cost to be accurate, only 6 bits of security from memory access costs are required for Kyber512 to meet category 1, so in this case Kyber512 would meet category 1 even if the NTRU and NTRUprime submission significantly overestimate the cost of memory access in lattice sieving algorithms. Further, since about 5 of the 14 claimed bits of security by Matzov involved speedups to local computations in AllPairSearch (as described by section 6 of the MATZOV paper), it is likely that Kyber512 would not be brought below category 1 by the MATZOV attack, as long as state of the art lattice cryptanalyses prior to the MATZOV paper were bottlenecked by memory at all. However, we acknowledge there is some additional uncertainty in the exact complexity of the MATZOV attack (and all other sieving-based lattice attacks) due to the known-unknowns Dan alludes to (described with quantitative estimates in section 5.3 of the Kyber spec.) Nonetheless, even taking the most paranoid values for these known-unknowns (16 bits of security loss), the cost of memory access and/or algorithmically making memory access local, would still need to be less than what both the NTRU and NTRUPrime submissions assume. The low end estimate of approximately 20 bits (from the NTRU submission) is based on a conjecture by Ducas that a fully local implementation of the BGJ1 sieving algorithm is possible. So, in the case that all known-unknowns take on the most paranoid values, this would either require a sieving algorithm with local memory access that is much better than any such published algorithm, and in fact better than any that has been conjectured (at least as far as we are aware), or it would require the approximately 40 bits of additional security quoted as the "real cost of memory access" by the NTRUprime submission to be a massive overestimate. In any event, a lot of things would have to go wrong simultaneously to push the real-world classical cost of known attacks against Kyber512 below category 1, which is why we don't think it's terribly likely.

As a final note, known quantum speedups for lattice sieving are much less effective than Grover’s algorithm for brute force key search, so in the likely scenario where the limiting attack on AES128 is Grover’s algorithm, this would further increase the security margin of Kyber512 over AES128 in practice.

Ray Perlner (NIST PQC)

-----Original Message-----
From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> On Behalf Of D. J. Bernstein
Sent: Saturday, December 3, 2022 11:11 AM
To: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms

'dustin…@nist.gov' via pqc-forum writes:
> What we meant was: There are various possible improvements in
> cryptanalysis, including: (i) optimization of lattice algorithms to
> improve the locality of memory access, (ii) building memory hardware
> that reduces the cost of non-local memory access, and (iii) other
> (more fundamental) improvements in the attacks. It seems unlikely that
> (i) and (ii) alone will be sufficient to cause Kyber-512 to fall below
> category 1 security, in realistic models of security that take these costs into account.

Thanks for the clarification.

In the interests of transparency and falsifiability, can NIST please explain how it arrived at the above likelihood assessment? The pieces sound like

(1) thinking that known Kyber-512 attacks haven't dropped far below single-target AES-128 key search (2^143) in gate counts,

(2) thinking that the costs of memory gain many more bits of security against those Kyber-512 attacks,

(3) thinking that improvements in algorithmic memory locality are unlikely to reduce Kyber-512's security level by much, and

(4) thinking that improvements in physical memory hardware (compared to computation) are unlikely to reduce Kyber-512's security level by much,

but I don't see any documentation of how NIST arrived at any of these individual conclusions, never mind evaluating the likelihood of error in the overall conclusion.

Of course, I checked the latest Kyber document from the Kyber team:

https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fpq-crystals.org%2Fkyber%2Fdata%2Fkyber-specification-round3-20210804.pdf&amp;data=05%7C01%7Cyi-kai.liu%40nist.gov%7C5698dfa237584e404afe08dad8a3fd48%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638060496089215178%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=suRdOS%2Btt3PM5DO%2BQiCiaAIfusZaoTqvEl5iWb%2BqUNU%3D&amp;reserved=0

This document addresses #1 but is obviously outdated, estimating 151 where typical estimates today are in the 130s. Which of the newer attacks and analyses is NIST taking into account? Is NIST's evaluation of #1 accounting for "known unknowns", and, if so, how?

As for the costs of memory, the 2021 Kyber document makes a brief, unquantified claim that 135 wouldn't be "catastrophic, in particular given the massive memory requirements that are ignored in the gate-count metric". Is this somehow the basis of NIST arriving at #2 or #3 or #4?

---D. J. Bernstein

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/20221203161126.150226.qmail%40cr.yp.to.

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/DM6PR09MB4855D376FB3C46592CE9A86E9C1A9%40DM6PR09MB4855.namprd09.prod.outlook.com.

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, December 07, 2022 09:10:52 PM ET
Attachments: smime.p7m

'Perlner, Ray A. (Fed)' via pqc-forum writes:

> We can elaborate a little bit further on our reasoning leading to our
> current assessment that Kyber512 likely meets NIST category I

Thanks. I've read through, and I have a much more specific clarification question to make sure I understand the underlying calculations. Within the space of scenarios reviewed, if we take the particular scenario of

(1) assuming accuracy of 2^137 from the most recent attack paper taken into account (Matzov) regarding the number of "gates",

(2) assuming this isn't affected by the "known unknowns", and

(3) assuming accuracy of the RAM-cost model in the NTRU Prime documentation,

then am I correctly gathering that you're calculating the Kyber-512 security level as 2^177 (i.e., 34 bits of security margin compared to 2^143 for AES-128), where this 177 comes from the above 137 plus 40, where 40 comes from 169 minus 129 on page 103 of the NTRU Prime documentation, specifically "real" minus "free" for pre-quantum sieving for sntrup653?

---D. J. Bernstein

--

From: Kampanakis, Panos <kpanos@amazon.com> via pqc-forum <pqc-forum@list.nist.gov>
To: Joost Renes <joost.renes@nxp.com>, pqc-forum <pqc-forum@list.nist.gov>
CC: Moody, Dustin (Fed) <dustin.moody@nist.gov>
Subject: RE: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Monday, December 12, 2022 03:24:37 PM ET

Hi Joost,

I think you are suggesting to support a mu calculation in Dilithium based on SHA-256 instead of SHAKE-256 because it will be much faster for bigger messages. A similar argument is made for SPHINCS+'s SHA256 versions. You are also suggesting that SHA-256 should still be supported as a message digest function in the digest-then-sign usecases like the automotive image signing because of the performance benefits. But if you digest the message beforehand, then the hash function used in calculating mu in Dilithium or Hmsg in SPHINCS+ will not make much difference because the message input to the signature is very small.

I want to clarify. Are you saying that if SHA-256 is used in the mu or Hmsg calculations, then these usecases could stop using digest-then-sign? Or are you suggesting to add SHA-256 support for the mu calculation in Dilithium for the general case where the message can be relatively big?

From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> **On Behalf Of** Joost Renes
Sent: Monday, December 12, 2022 11:30 AM
To: pqc-forum <pqc-forum@list.nist.gov>
Cc: Moody, Dustin (Fed) <dustin.moody@nist.gov>
Subject: RE: [EXTERNAL][EXT] [pqc-forum] Parameter selection for the selected algorithms

Hi Dustin,

We would like to put forward the option of allowing SHA-2 for the Dilithium message hash mu = H(tr || M).

We are generally in favor of allowing NIST primitives which are already widely deployed and have seen wide adoption in (embedded) systems, such as AES and SHA-2.

While we agree that the simplicity of a single symmetric primitive (SHAKE) is preferable from a design perspective, the decision of a user for adoption and migration towards PQC ultimately is determined by the impact on their system.

There can be a significant difference in performance between HW-accelerated AES and SHA-2, and SHAKE which is mostly not yet available in HW.

Indeed this is a problem that would resolve itself in the medium to long term, but we believe many systems will significantly benefit from using a SHA-2 / AESversion where it might make the difference between PQC being feasible or not.

Having that said, we would like to particularly emphasize the message hash itself.

In most performance benchmarks of Dilithium this is barely taken into account: e.g., the specification benchmarks 64-byte messages for AVX2 [1] while pqm4 uses 59-byte messages [2].

However, for many applications the message hash may be a significant part of the performance.

Our investigations for PQC enabled secure boot on one of NXP's automotive platforms led to the conclusion that with SHA-3 in software the hash of a 128 KiB image takes ~150ms while the remainder only requires ~16.7ms [3, Table 1].

Realistic images for automotive applications can even be megabytes large, making the message hash the dominant cost by far.

For comparison: with HW-accelerated SHA-2 the 128 KiB message hash takes only 0.2ms.

This is not an isolated case; SHA-2 is much more widely adopted in hardware than SHA-3.

Again, to push for PQC adoption it is very helpful to allow this improvement in performance.

This is a very similar argument as for allowing SHA-2 as a parameter set for SPHINCS+.

Finally, this only involves the digest creation which is often done external to the signing/ verification (as also discussed in the past on this mailing list) so would have little impact on the complexity of Dilithium itself.

We look forward to hearing your thoughts (and anyone else's!).

Kind regards,

NXP PQC team

[1] https://github.com/pq-crystals/dilithium/blob/master/ref/test/test_speed.c

[2] https://github.com/mupq/mupq/blob/3b48fa5aff6f5921df5b3444450281daca6d21d1/crypto_sign/speed.c

[3] https://eprint.iacr.org/2022/635.pdf

From: 'Moody, Dustin (Fed)' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Wednesday, November 30, 2022 1:26 PM
To: pqc-forum <pqc-forum@list.nist.gov>
Subject: [EXT] [pqc-forum] Parameter selection for the selected algorithms

Caution: EXT Email

Dear all,

NIST has been thinking a lot about which parameter sets to include as part of the standards for the selected algorithms. We wanted to share our current view.

Dilithium

We are planning to include parameter sets corresponding to security categories 2, 3, and 5. We are not planning to include the AES variant.

Falcon

We are planning to include parameter sets corresponding to security categories 1 and 5.

SPHINCS+

We plan to include parameter sets for security categories 1, 3, and 5. We plan to include the simple (and NOT the robust) version. We also plan to include both the fast and small versions. Allowed hash functions will be SHAKE and SHA-2 (SHA-256 for category 1 and a mix of SHA-512 and SHA-256 for categories 3 and 5).

Kyber

We are planning on including parameter sets for categories 1, 3, and 5, though we would highly recommend the category 3 parameter set as the default option. We are not planning on including the 90s variant.

The decision for the category 1 parameter set has been more difficult. There have been extensive discussions if and in what metrics this parameter set achieves the same security as AES-128. It is clear that in the gate-count metric it is a very close call and that in this metric the pre-quantum security of Kyber-512 may be a few bits below the one of AES-128. However, the best known attacks against Kyber-512 require huge amounts of memory and the real attack cost will need to take the cost of (access to) memory into account. This cost is not easy to calculate, as it depends on the memory access patterns of the lattice algorithms used for cryptanalysis, as well as the physical properties of the memory hardware. Nonetheless, barring major improvements in cryptanalysis, it seems unlikely that the cost of memory access will ever become small enough to cause Kyber-512 to fall below category 1 security, in realistic models of security that take these costs into account. We acknowledge there can be different views on our current view to include Kyber-512.

As a point of clarification: in this email, we refer to parameter sets based on the claimed security strength category where those parameter sets are most recently specified, irrespective of whether those parameter sets actually meet their claimed security level. That said, our current assessment is that, when realistic memory access costs of known attacks are taken into account, all the parameter sets we plan to standardize do, in fact, meet their claimed security strength categories.

NIST PQC team

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/SA1PR09MB8669BA6FE29FFC5D68C04E3CE5159%40SA1PR09MB8669.namprd09.prod.outlook.com.

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/AM9PR04MB8161AD39A8403821E0C79400FFE29%40AM9PR04MB8161.eurprd04.prod.outlook.com.

From: Markku-Juhani O. Saarinen <mjos.crypto@gmail.com> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
CC: joost.renes <joost.renes@nxp.com>, dustin...@nist.gov <dustin.moody@nist.gov>, Kampanakis, Panos <kpanos@amazon.com>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Monday, December 12, 2022 04:25:37 PM ET

On Monday, December 12, 2022 at 5:31:47 PM UTC+1 joost.renes wrote:
> We would like to put forward the option of allowing SHA-2 for the Dilithium message hash mu = H(tr || M).

On Monday, December 12, 2022 at 9:24:06 PM UTC+1 Kampanakis, Panos wrote:
> I think you are suggesting to support a mu calculation in Dilithium based on SHA-256 instead of SHAKE-256 because it will be much faster for bigger messages.

Hello All,

What was suggested was the computation of mu as "mu = H(tr || M)." with SHA-2. I support this option (and it should be an option), but the instance of H should be SHA-512, not SHA-256.

Rationale: SHA-2 algorithms are widely available and fast (there are effectively two distinct SHA-2 algorithms; "32-bit" SHA-224/256 and "64-bit" SHA-384/512 -- the latter is faster on PCs), and it is good to be able to use an "external" SHA-2 hashing engine in hardware use cases.

- Recall that mu is 512 bits; currently 64 bytes from SHAKE256. The "256" in SHAKE256 refers to the "256-bit" security level; it's an XOF that can produce an output sequence of arbitrary length. Internally SHAKE256 has a 512-bit chaining capacity (just like SHA-512).
- For internal hash/XOF operations, I would prefer to stick to Keccak-based procedures. Otherwise, hardware modules (especially side-channel secure ones) will become much more complex to implement as they would need to support many internal options ("mu" is effectively external to the signature generation and verification logic). A masked Keccak is relatively large but has many redeeming qualities when compared to masked versions of SHA-2.
- Also, we should stay with a single definition of "tr," which is a 256-bit SHAKE256 hash of the public key tr=H(rho || t1). It is also a part of the secret key as Dilithium doesn't need the full

public key to perform the private key operation, just the "tr" prefix.

On Monday, December 12, 2022 at 10:17:16 PM UTC+1 joost.renes wrote:
> In particular because Dilithium does consider the digest computation part of the specification: Instead of double-digest computations, we would prefer the option of explicitly allowing SHA2.

Yes, double-digest computations should be avoided. Some things I've seen explicitly break the security proofs of Dilithium (e.g., those that just re-hash and sign a plain hash H(m), making the scheme dependant on plain collision resistance again.)

Cheers,
- markku

Dr. Markku-Juhani O. Saarinen

On Monday, December 12, 2022 at 10:17:16 PM UTC+1 joost.renes wrote:

> Hi Panos,
>
> Indeed digest-then-sign is a solution, as we also explored in [3].
>
> But since using SHA2 will be beneficial on such a wide variety of system, we rather prefer it to be included in the standard rather than built on top in an ad hoc fashion.
>
> In particular because Dilithium does consider the digest computation part of the specification: Instead of double-digest computations, we would prefer the option of explicitly allowing SHA2.
>
> Kind regards,
>
> Joost

From: Kampanakis, Panos <kpa...@amazon.com>
Sent: Monday, December 12, 2022 9:24 PM
To: Joost Renes <joost...@nxp.com>; pqc-forum <pqc-...@list.nist.gov>
Cc: Moody, Dustin (Fed) <dustin...@nist.gov>
Subject: RE: [EXT] [pqc-forum] Parameter selection for the selected algorithms

Caution: EXT Email

Hi Joost,

I think you are suggesting to support a mu calculation in Dilithium based on SHA-256 instead of SHAKE-256 because it will be much faster for bigger messages. A similar argument is made for SPHINCS+'s SHA256 versions. You are also suggesting that SHA-256 should still be supported as a message digest function in the digest-then-sign usecases like the automotive image signing because of the performance benefits. But if you digest the message beforehand, then the hash function used in calculating mu in Dilithium or Hmsg in SPHINCS+ will not make much difference because the message input to the signature is very small.

I want to clarify. Are you saying that if SHA-256 is used in the mu or Hmsg calculations, then these usecases could stop using digest-then-sign? Or are you suggesting to add SHA-256 support for the mu calculation in Dilithium for the general case where the message can be relatively big?

From:pqc-...@list.nist.gov <pqc-...@list.nist.gov> **On Behalf Of** Joost Renes
Sent: Monday, December 12, 2022 11:30 AM
To: pqc-forum <pqc-...@list.nist.gov>
Cc: Moody, Dustin (Fed) <dustin...@nist.gov>
Subject: RE: [EXTERNAL][EXT] [pqc-forum] Parameter selection for the selected algorithms

Hi Dustin,

We would like to put forward the option of allowing SHA-2 for the Dilithium message hash mu = H(tr || M).

We are generally in favor of allowing NIST primitives which are already widely deployed and have seen wide adoption in (embedded) systems, such as AES and SHA-2.

> While we agree that the simplicity of a single symmetric primitive (SHAKE) is preferable from a design perspective, the decision of a user for adoption and migration towards PQC ultimately is determined by the impact on their system.
>
> There can be a significant difference in performance between HW-accelerated AES and SHA-2, and SHAKE which is mostly not yet available in HW.
>
> Indeed this is a problem that would resolve itself in the medium to long term, but we believe many systems will significantly benefit from using a SHA-2 / AESversion where it might make the difference between PQC being feasible or not.
>
> Having that said, we would like to particularly emphasize the message hash itself.
>
> In most performance benchmarks of Dilithium this is barely taken into account: e.g., the specification benchmarks 64-byte messages for AVX2 [1] while pqm4 uses 59-byte messages [2].
>
> However, for many applications the message hash may be a significant part of the performance.
>
> Our investigations for PQC enabled secure boot on one of NXP’s automotive platforms led to the conclusion that with SHA-3 in software the hash of a 128 KiB image takes ~150ms while the remainder only requires ~16.7ms [3, Table 1].
>
> Realistic images for automotive applications can even be megabytes large, making the message hash the dominant cost by far.
>
> For comparison: with HW-accelerated SHA-2 the 128 KiB message hash takes only 0.2ms.
>
> This is not an isolated case; SHA-2 is much more widely adopted in hardware than SHA-3.
>
> Again, to push for PQC adoption it is very helpful to allow this improvement in performance.
>
> This is a very similar argument as for allowing SHA-2 as a parameter set for SPHINCS+.
>
> Finally, this only involves the digest creation which is often done external to the signing/verification (as also discussed in the past on this mailing list) so would have little impact on the complexity of Dilithium itself.
>
> We look forward to hearing your thoughts (and anyone else’s!).
>
> Kind regards,

NXP PQC team

[1] https://github.com/pq-crystals/dilithium/blob/master/ref/test/test_speed.c

[2] https://github.com/mupq/mupq/blob/3b48fa5aff6f5921df5b3444450281daca6d21d1/crypto_sign/speed.c

[3] https://eprint.iacr.org/2022/635.pdf

From: 'Moody, Dustin (Fed)' via pqc-forum <pqc-...@list.nist.gov>
Sent: Wednesday, November 30, 2022 1:26 PM
To: pqc-forum <pqc-...@list.nist.gov>
Subject: [EXT] [pqc-forum] Parameter selection for the selected algorithms

Caution: EXT Email

Dear all,

NIST has been thinking a lot about which parameter sets to include as part of the standards for the selected algorithms. We wanted to share our current view.

Dilithium

We are planning to include parameter sets corresponding to security categories 2, 3, and 5. We are not planning to include the AES variant.

Falcon

We are planning to include parameter sets corresponding to security categories 1 and 5.

SPHINCS+

We plan to include parameter sets for security categories 1, 3, and 5. We plan to include the simple (and NOT the robust) version. We also plan to include both the fast and small versions. Allowed hash functions will be SHAKE and SHA-2 (SHA-256 for category 1 and a mix of SHA-512 and SHA-256 for categories 3 and 5).

Kyber

We are planning on including parameter sets for categories 1, 3, and 5, though we would highly recommend the category 3 parameter set as the default option. We are not planning on including the 90s variant.

> The decision for the category 1 parameter set has been more difficult. There have been extensive discussions if and in what metrics this parameter set achieves the same security as AES-128. It is clear that in the gate-count metric it is a very close call and that in this metric the pre-quantum security of Kyber-512 may be a few bits below the one of AES-128. However, the best known attacks against Kyber-512 require huge amounts of memory and the real attack cost will need to take the cost of (access to) memory into account. This cost is not easy to calculate, as it depends on the memory access patterns of the lattice algorithms used for cryptanalysis, as well as the physical properties of the memory hardware. Nonetheless, barring major improvements in cryptanalysis, it seems unlikely that the cost of memory access will ever become small enough to cause Kyber-512 to fall below category 1 security, in realistic models of security that take these costs into account. We acknowledge there can be different views on our current view to include Kyber-512.
>
> As a point of clarification: in this email, we refer to parameter sets based on the claimed security strength category where those parameter sets are most recently specified, irrespective of whether those parameter sets actually meet their claimed security level. That said, our current assessment is that, when realistic memory access costs of known attacks are taken into account, all the parameter sets we plan to standardize do, in fact, meet their claimed security strength categories.
>
> NIST PQC team
>
> --
> You received this message because you are subscribed to the Google Groups "pqc-forum" group.
> To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+...@list.nist.gov.
> To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/SA1PR09MB8669BA6FE29FFC5D68C04E3CE5159%40SA1PR09MB8669.namprd09.prod.outlook.com.
>
> --
> You received this message because you are subscribed to the Google Groups "pqc-forum" group.
> To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+...@list.nist.gov.

> To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/AM9PR04MB8161AD39A8403821E0C79400FFE29%40AM9PR04MB8161.eurprd04.prod.outlook.com.

--

From: Peter Schwabe <peter@cryptojedi.org> via pqc-forum@list.nist.gov
To: Joost Renes <joost.renes@nxp.com>
CC: pqc-forum <pqc-forum@list.nist.gov>, Moody, Dustin (Fed) <dustin.moody@nist.gov>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, December 13, 2022 09:02:26 AM ET

Joost Renes <joost.renes@nxp.com> wrote:

> Hi Dustin,

Hi Joost,

> We would like to put forward the option of allowing SHA-2 for the Dilithium
> message hash mu = H(tr || M).

Would it alternatively work for you to standardize also
"HashedDilithium", i.e., a version that takes the message M, first
computes M' = H(M) with any FIPS-certifiable hash function H at suitable
security level and then run Dilithium.Sign(M',sk)?

All the best,

Peter

--

From: Blumenthal, Uri - 0553 - MITLL <uri@ll.mit.edu> via pqc-forum@list.nist.gov
To: Peter Schwabe <peter@cryptojedi.org>, Joost Renes <joost.renes@nxp.com>
CC: pqc-forum <pqc-forum@list.nist.gov>, Moody, Dustin (Fed) <dustin.moody@nist.gov>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, December 13, 2022 10:43:43 AM ET
Attachments: smime.p7m

Standardizing "HashedDilithium" makes sense to me.

Thanks

--

V/R,

Uri

On 12/13/22, 09:02, "pqc-forum@list.nist.gov on behalf of Peter Schwabe" <pqc-forum@list.nist.gov on behalf of peter@cryptojedi.org> wrote:

Joost Renes <joost.renes@nxp.com> wrote:

> Hi Dustin,

Hi Joost,

> We would like to put forward the option of allowing SHA-2 for the Dilithium
> message hash mu = H(tr || M).

Would it alternatively work for you to standardize also "HashedDilithium", i.e., a version that takes the message M, first computes M' = H(M) with any FIPS-certifiable hash function H at suitable security level and then run Dilithium.Sign(M',sk)?

All the best,

Peter

--

You received this message because you are subscribed to the Google Groups "pqc-forum" group.

```
    To unsubscribe from this group and stop receiving emails from it, send an email
to pqc-forum+unsubscribe@list.nist.gov.
    To view this discussion on the web visit https://groups.google.com/a/
list.nist.gov/d/msgid/pqc-forum/Y5iFp7Fen56Lr7Nk%40disp3269.
```

```
--
You received this message because you are subscribed to the Google Groups "pqc-forum"
group.
To unsubscribe from this group and stop receiving emails from it, send an email to
pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/
msgid/pqc-forum/5EBFA580-EFDA-49BB-87C9-14D6F1836004%40ll.mit.edu.
```

From: Markku-Juhani O. Saarinen <mjos.crypto@gmail.com> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
CC: joost.renes <joost.renes@nxp.com>, pqc-forum <pqc-forum@list.nist.gov>, dustin...@nist.gov <dustin.moody@nist.gov>, Peter Schwabe <peter@cryptojedi.org>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, December 13, 2022 01:45:07 PM ET

On Mon, Dec 12, 2022 at 5:31 PM Joost Renes wrote:
> We would like to put forward the option of allowing SHA-2 for the
> Dilithium message hash mu = H(tr || M).

On Tue, Dec 13, 2022 at 3:01 PM Peter Schwabe wrote:
> Would it alternatively work for you to standardize also
> "HashedDilithium", i.e., a version that takes the message M, first
> computes M' = H(M) with any FIPS-certifiable hash function H at suitable
> security level and then run Dilithium.Sign(M',sk)?

Hi Peter and all,

Allowing H(M) for Dilithium2 is potentially helpful for shortest-term transitioning pure consumer-level systems, but something different would be needed for security Level 5, which is the only version allowed in higher-security applications.

Concretely: While the definition of "suitable security level" hash is reasonably clear for Dilithium2, what would it be for Dilithium5 -- especially if the "tr" prefix is removed and collision resistance is assumed?

- Background note to those looking at the specs: Dilithium was already changed once during Round 3 to meet the Level 5 PQ requirement; the Round 3 submission document (v3.0 / 20201001) on the NIST website uses a weaker 384-bit "mu" length parameter; at the suggestion of external cryptanalysis this was increased to 512 bits for Dilithium v3.1 / 20210208 which is only available at the Dilithium designer's web site. https://pq-crystals.org/dilithium/resources.shtml

- Removing the "tr" prefix from the initial hash opens the scheme to generic collision attacks rather than attacks targeted at a specific public key/identity. The classical security of SHA2 against these types of attacks is traditionally upper bound at the "256-bit" security level. Arguments can be made that the quantum security level could be less, and hence H(M) with

SHA2-512 would not meet PQ Security Level 5.

- Plain (un-prefixed) H(M) with SHA3-512 might be good enough for PQC Level 5 (SHA3 does not have a 256-bit classical attack design upper bound like SHA2), but additional quantum security analysis would be needed. Note that SHA3-512 requires 89% more permutations to process M than SHAKE256; hence it runs at almost half the speed (per kilobyte of data signed/verified) compared to the current Dilithium5 v3.1 design.

- On "FIPS-certifiable:" Having had some exposure to FIPS certification, I would try to avoid this language here in particular. Even though the weakest component sets a system's security level, FIPS currently allows any combination of hashes and signature algorithms as long as all components have a classical security level of 112 bits. Typical FIPS-certified security stacks and signature/PKI solutions have a 128-bit overall classical security level, even if some individual components are designed for higher security levels ("256-bit TLS" is generally marketing language only.)

Cheers,

- markku

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

On Tuesday, December 13, 2022 at 5:35:17 PM UTC+1 joost.renes wrote:

> Hi Peter,
>
> That would alleviate the performance concerns, so yes we would be in favor
> of having such a version in the standard.
> Could you elaborate on why this is preferable over allowing this instance of
> H to be a FIPS-certifiable function itself?
> It seems to me more natural than chaining the digests as would be done in
> hashedDilithium.
>
> Kind regards,
> Joost

> -----Original Message-----
> From: Peter Schwabe <pe...@cryptojedi.org>
> Sent: Tuesday, December 13, 2022 3:01 PM
> To: Joost Renes <joost...@nxp.com>
> Cc: pqc-forum <pqc-...@list.nist.gov>; Moody, Dustin (Fed)
> <dustin...@nist.gov>
> Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected
> algorithms
>
> Caution: EXT Email
>
> Joost Renes <joost...@nxp.com> wrote:
> > Hi Dustin,
>
> Hi Joost,
>
> > We would like to put forward the option of allowing SHA-2 for the
> > Dilithium message hash mu = H(tr || M).
>
> Would it alternatively work for you to standardize also "HashedDilithium",
> i.e., a version that takes the message M, first computes M' = H(M) with any
> FIPS-certifiable hash function H at suitable security level and then run
> Dilithium.Sign(M',sk)?
>
> All the best,
>
> Peter

--

You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/f08bf458-262d-4bbc-8afb-ab687a18bb1en%40list.nist.gov.

From: Markku-Juhani O. Saarinen <mjos.crypto@gmail.com> via pqc-forum@list.nist.gov
To: pqc-forum <pqc-forum@list.nist.gov>
CC: joost.renes <joost.renes@nxp.com>, pqc-forum <pqc-forum@list.nist.gov>, dustin...@nist.gov <dustin.moody@nist.gov>, Peter Schwabe <peter@cryptojedi.org>, Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, December 13, 2022 06:16:09 PM ET

On Tuesday, December 13, 2022 at 9:24:30 PM UTC+1 joost.renes wrote:

> Hi Markku,
>
> Would this not be solved by setting M' = H(tr||M) and running Dilithium.Sign(M',sk) as proposed by Peter?
>
> That would have negligible overhead since tr is computed anyway.
>
> (Though I am not convinced yet it is necessary, see questions/comments below.)

Yes, the original M'=H(tr || M) would be preferable M'=H(M). As you note, it has a negligible performance penalty, and it does provide some additional security against collision attacks.

Most modern signature schemes (EdDSA, XMSS, LMS/HSS, Falcon, Sphincs+,..) have some variant of a hash target prefix (represented here by "tr"). Prominent 90s pedigree schemes such as PCKS #1 v1.5 RSA and (EC)DSA didn't have it, and hence there is resistance in some quarters to it due to API and protocol limitations. However, I think the additional security properties are worth the migration effort.

> "Arguments can be made that the quantum security level could be less, and hence H(M) with SHA2-512 would not meet PQ Security Level 5."
>
> I don't really understand this point: it might have lower quantum security, but the same can be argued about AES256. The definition of the NIST security levels says SHA256 (L2) > AES128 (L1) and SHA384 (L4) > AES192 (L3). What is special about SHA512 that we do not have SHA512 > AES256 (L5)?

I see your point; an assumption has always been made that SHA256 > AES128 which sort of implies SHA512 > AES256. I agree that 512-bit collision resistance should be sufficient for Level 5 (some quantum algorithm theory person can comment in detail, but my understanding is that Grover's attack is relatively so much more effective than any PQ collision search.)

Cheers,

- markku

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

From: Markku-Juhani O. Saarinen <mjos....@gmail.com>
Sent: Tuesday, December 13, 2022 7:45 PM
To: pqc-forum <pqc-...@list.nist.gov>

Cc: Joost Renes <joost...@nxp.com>; pqc-forum <pqc-...@list.nist.gov>; dustin...@nist.gov <dustin...@nist.gov>; Peter Schwabe <pe...@cryptojedi.org>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms

Caution: EXT Email

On Mon, Dec 12, 2022 at 5:31 PM Joost Renes <joost...@nxp.com> wrote:
> We would like to put forward the option of allowing SHA-2 for the
> Dilithium message hash mu = H(tr || M).

On Tue, Dec 13, 2022 at 3:01 PM Peter Schwabe <pe...@cryptojedi.org> wrote:
> Would it alternatively work for you to standardize also
> "HashedDilithium", i.e., a version that takes the message M, first
> computes M' = H(M) with any FIPS-certifiable hash function H at suitable
> security level and then run Dilithium.Sign(M',sk)?

Hi Peter and all,

Allowing H(M) for Dilithium2 is potentially helpful for shortest-term transitioning pure consumer-level systems, but something different would be needed for security Level 5, which is the only version allowed in higher-security applications.

Concretely: While the definition of "suitable security level" hash is reasonably clear for Dilithium2, what would it be for Dilithium5 -- especially if the "tr" prefix is removed and collision resistance is assumed?

- Background note to those looking at the specs: Dilithium was already changed once during Round 3 to meet the Level 5 PQ requirement; the Round 3 submission document (v3.0 / 20201001) on the NIST website uses a weaker 384-bit "mu" length parameter; at the

> suggestion of external cryptanalysis this was increased to 512 bits for Dilithium v3.1 / 20210208 which is only available at the Dilithium designer's web site. https://pq-crystals.org/dilithium/resources.shtml
>
> - Removing the "tr" prefix from the initial hash opens the scheme to generic collision attacks rather than attacks targeted at a specific public key/identity. The classical security of SHA2 against these types of attacks is traditionally upper bound at the "256-bit" security level. Arguments can be made that the quantum security level could be less, and hence H(M) with SHA2-512 would not meet PQ Security Level 5.
>
> - Plain (un-prefixed) H(M) with SHA3-512 might be good enough for PQC Level 5 (SHA3 does not have a 256-bit classical attack design upper bound like SHA2), but additional quantum security analysis would be needed. Note that SHA3-512 requires 89% more permutations to process M than SHAKE256; hence it runs at almost half the speed (per kilobyte of data signed/verified) compared to the current Dilithium5 v3.1 design.
>
> - On "FIPS-certifiable:" Having had some exposure to FIPS certification, I would try to avoid this language here in particular. Even though the weakest component sets a system's security level, FIPS currently allows any combination of hashes and signature algorithms as long as all components have a classical security level of 112 bits. Typical FIPS-certified security stacks and signature/PKI solutions have a 128-bit overall classical security level, even if some individual components are designed for higher security levels ("256-bit TLS" is generally marketing language only.)
>
> Cheers,
>
> - markku
>
> Dr. Markku-Juhani O. Saarinen <mj...@iki.fi>
>
> On Tuesday, December 13, 2022 at 5:35:17 PM UTC+1 joost.renes wrote:
>
>> Hi Peter,
>>
>> That would alleviate the performance concerns, so yes we would be in favor
>> of having such a version in the standard.
>> Could you elaborate on why this is preferable over allowing this instance of
>> H to be a FIPS-certifiable function itself?
>> It seems to me more natural than chaining the digests as would be done in
>> hashedDilithium.

> > Kind regards,
> > Joost
> >
> > -----Original Message-----
> > From: Peter Schwabe <pe...@cryptojedi.org>
> > Sent: Tuesday, December 13, 2022 3:01 PM
> > To: Joost Renes <joost...@nxp.com>
> > Cc: pqc-forum <pqc-...@list.nist.gov>; Moody, Dustin (Fed)
> > <dustin...@nist.gov>
> > Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected
> > algorithms
> >
> > Caution: EXT Email
> >
> > Joost Renes <joost...@nxp.com> wrote:
> > > Hi Dustin,
> >
> > Hi Joost,
> >
> > > We would like to put forward the option of allowing SHA-2 for the
> > > Dilithium message hash mu = H(tr || M).
> >
> > Would it alternatively work for you to standardize also "HashedDilithium",
> > i.e., a version that takes the message M, first computes M' = H(M) with any
> > FIPS-certifiable hash function H at suitable security level and then run
> > Dilithium.Sign(M',sk)?
> >
> > All the best,
> >
> > Peter

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.

To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/a0917aa3-1ceb-44be-bcf4-14426cec350dn%40list.nist.gov.

From: Kampanakis, Panos <kpanos@amazon.com> via pqc-forum <pqc-forum@list.nist.gov>
To: Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>, pqc-forum <pqc-forum@list.nist.gov>
CC: joost.renes <joost.renes@nxp.com>, dustin...@nist.gov <dustin.moody@nist.gov>, Peter Schwabe <peter@cryptojedi.org>
Subject: RE: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, December 13, 2022 09:41:00 PM ET

Another argument against the spec including a prehash version is history. Defining PureEdDSA and PrehashEdDSA didn't help interop and in the end only PureEdDSA got used in most use-cases (eg. TLS, IKEv2, SSH, X509, CMS).

If a use-case needs to prehash the message, it can do so. PKCS#11, for example, offers a PrehashEdDSA mechanism for cases where the message is so big that it can't be cached in the signer or the verifier. Specifically for Dilithium, this is not even necessary because H(tr || M) can be calculated by the signer/verifier without a problem by using the PKCS#11 incremental API (C_SignUpdate/ C_VerifyUpdate).

From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> **On Behalf Of** Markku-Juhani O. Saarinen
Sent: Tuesday, December 13, 2022 6:16 PM
To: pqc-forum <pqc-forum@list.nist.gov>
Cc: joost.renes <joost.renes@nxp.com>; pqc-forum <pqc-forum@list.nist.gov>; dustin...@nist.gov <dustin.moody@nist.gov>; Peter Schwabe <peter@cryptojedi.org>; Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
Subject: RE: [EXTERNAL][EXT] [pqc-forum] Parameter selection for the selected algorithms

CAUTION: This email originated from outside of the organization. Do not click links or open attachments unless you can confirm the sender and know the content is safe.

On Tuesday, December 13, 2022 at 9:24:30 PM UTC+1 joost.renes wrote:

> Hi Markku,
>
> Would this not be solved by setting M’ = H(tr||M) and running Dilithium.Sign(M’,sk) as proposed by Peter?
>
> That would have negligible overhead since tr is computed anyway.
>
> (Though I am not convinced yet it is necessary, see questions/comments below.)

Yes, the original M'=H(tr || M) would be preferable M'=H(M). As you note, it has a negligible performance penalty, and it does provide some additional security against collision attacks.

Most modern signature schemes (EdDSA, XMSS, LMS/HSS, Falcon, Sphincs+,..) have some variant of a hash target prefix (represented here by "tr"). Prominent 90s pedigree schemes such as PCKS #1 v1.5 RSA and (EC)DSA didn't have it, and hence there is resistance in some quarters to it due to API and protocol limitations. However, I think the additional security properties are worth the migration effort.

> “Arguments can be made that the quantum security level could be less, and hence H(M) with SHA2-512 would not meet PQ Security Level 5.”
>
> I don’t really understand this point: it might have lower quantum security, but the same can be argued about AES256. The definition of the NIST security levels says SHA256 (L2) > AES128 (L1) and SHA384 (L4) > AES192 (L3). What is special about SHA512 that we do not have SHA512 > AES256 (L5)?

I see your point; an assumption has always been made that SHA256 > AES128 which sort of implies SHA512 > AES256. I agree that 512-bit collision resistance should be sufficient for Level 5 (some quantum algorithm theory person can comment in detail, but my understanding is that Grover's attack is relatively so much more effective than any PQ collision search.)

Cheers,

- markku

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

> **From:** Markku-Juhani O. Saarinen <mjos....@gmail.com>
> **Sent:** Tuesday, December 13, 2022 7:45 PM
> **To:** pqc-forum <pqc-...@list.nist.gov>
>
> **Cc:** Joost Renes <joost...@nxp.com>; pqc-forum <pqc-...@list.nist.gov>; dustin...@nist.gov <dustin...@nist.gov>; Peter Schwabe <pe...@cryptojedi.org>
> **Subject:** Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
>
> **Caution:** EXT Email
>
> On Mon, Dec 12, 2022 at 5:31 PM Joost Renes <joost...@nxp.com> wrote:
> > We would like to put forward the option of allowing SHA-2 for the
> > Dilithium message hash mu = H(tr || M).

On Tue, Dec 13, 2022 at 3:01 PM Peter Schwabe <pe...@cryptojedi.org> wrote:
> Would it alternatively work for you to standardize also
> "HashedDilithium", i.e., a version that takes the message M, first
> computes M' = H(M) with any FIPS-certifiable hash function H at suitable
> security level and then run Dilithium.Sign(M',sk)?

Hi Peter and all,

Allowing H(M) for Dilithium2 is potentially helpful for shortest-term transitioning pure consumer-level systems, but something different would be needed for security Level 5, which is the only version allowed in higher-security applications.

Concretely: While the definition of "suitable security level" hash is reasonably clear for Dilithium2, what would it be for Dilithium5 -- especially if the "tr" prefix is removed and collision resistance is assumed?

- Background note to those looking at the specs: Dilithium was already changed once during Round 3 to meet the Level 5 PQ requirement; the Round 3 submission document (v3.0 / 20201001) on the NIST website uses a weaker 384-bit "mu" length parameter; at the suggestion of external cryptanalysis this was increased to 512 bits for Dilithium v3.1 / 20210208 which is only available at the Dilithium designer's web site. https://pq-crystals.org/dilithium/resources.shtml

- Removing the "tr" prefix from the initial hash opens the scheme to generic collision attacks rather than attacks targeted at a specific public key/identity. The classical security of SHA2 against these types of attacks is traditionally upper bound at the "256-bit" security level. Arguments can be made that the quantum security level could be less, and hence H(M) with SHA2-512 would not meet PQ Security Level 5.

- Plain (un-prefixed) H(M) with SHA3-512 might be good enough for PQC Level 5 (SHA3 does not have a 256-bit classical attack design upper bound like SHA2), but additional quantum security analysis would be needed. Note that SHA3-512 requires 89% more permutations to process M than SHAKE256; hence it runs at almost half the speed (per kilobyte of data signed/verified) compared to the current Dilithium5 v3.1 design.

- On "FIPS-certifiable:" Having had some exposure to FIPS certification, I would try to avoid this language here in particular. Even though the weakest component sets a system's security level, FIPS currently allows any combination of hashes and signature algorithms as

> long as all components have a classical security level of 112 bits. Typical FIPS-certified security stacks and signature/PKI solutions have a 128-bit overall classical security level, even if some individual components are designed for higher security levels ("256-bit TLS" is generally marketing language only.)
>
> Cheers,
>
> - markku
>
> Dr. Markku-Juhani O. Saarinen <mj...@iki.fi>
>
> On Tuesday, December 13, 2022 at 5:35:17 PM UTC+1 joost.renes wrote:
>
>> Hi Peter,
>>
>> That would alleviate the performance concerns, so yes we would be in favor
>> of having such a version in the standard.
>> Could you elaborate on why this is preferable over allowing this instance of
>> H to be a FIPS-certifiable function itself?
>> It seems to me more natural than chaining the digests as would be done in
>> hashedDilithium.
>>
>> Kind regards,
>> Joost
>>
>> -----Original Message-----
>> From: Peter Schwabe <pe...@cryptojedi.org>
>> Sent: Tuesday, December 13, 2022 3:01 PM
>> To: Joost Renes <joost...@nxp.com>
>> Cc: pqc-forum <pqc-...@list.nist.gov>; Moody, Dustin (Fed)
>> <dustin...@nist.gov>
>> Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected
>> algorithms
>>
>> Caution: EXT Email
>>
>> Joost Renes <joost...@nxp.com> wrote:
>> > Hi Dustin,

> > Hi Joost,
> >
> > > We would like to put forward the option of allowing SHA-2 for the
> > > Dilithium message hash mu = H(tr || M).
> >
> > Would it alternatively work for you to standardize also "HashedDilithium",
> > i.e., a version that takes the message M, first computes M' = H(M) with any
> > FIPS-certifiable hash function H at suitable security level and then run
> > Dilithium.Sign(M',sk)?
> >
> > All the best,
> >
> > Peter

--

You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/a0917aa3-1ceb-44be-bcf4-14426cec350dn%40list.nist.gov.

From: John Mattsson <john.mattsson@ericsson.com> via pqc-forum <pqc-forum@list.nist.gov>
To: Kampanakis, Panos <kpanos@amazon.com>, Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>, pqc-forum <pqc-forum@list.nist.gov>
CC: joost.renes <joost.renes@nxp.com>, dustin...@nist.gov <dustin.moody@nist.gov>, Peter Schwabe <peter@cryptojedi.org>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, December 14, 2022 03:12:47 AM ET

Hi,

I agree. I think there is a lot to learn from the EdDSA standardization:

- Specifying PureEdDSA and PrehashEdDSA has so far just led to confusion. My preference would be to only have one version.

- Specifying only SHA-512 in Ed25519 was likely also a bad choice. EdDSA has not seen any use in Web Servers. It has also not been used in constrained IoT where the increased performance would have been very welcome. Many older devices only have hardware acceleration for SHA-256, and many newer IoT systems is looking at using Keccak only. For new algorithms, I think it make sense to go for Keccak only.
https://datatracker.ietf.org/meeting/100/materials/slides-100-t2trg-small-crypto-for-small-iot

- Only specifying a deterministic mode made EdDSA more or less unusable on IoT devices due to side-channel attacks. A purely randomized version works on some devices but not on many others. In general hedged signatures are needed (unless deterministic can be implemented in side-channel secure way).

Cheers,

John

From: 'Kampanakis, Panos' via pqc-forum <pqc-forum@list.nist.gov>
Date: Wednesday, 14 December 2022 at 03:41
To: Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>, pqc-forum <pqc-forum@list.nist.gov>
Cc: joost.renes <joost.renes@nxp.com>, dustin...@nist.gov <dustin.moody@nist.gov>, Peter

Schwabe <peter@cryptojedi.org>
Subject: RE: [EXT] [pqc-forum] Parameter selection for the selected algorithms

Another argument against the spec including a prehash version is history. Defining PureEdDSA and PrehashEdDSA didn't help interop and in the end only PureEdDSA got used in most use-cases (eg. TLS, IKEv2, SSH, X509, CMS).

If a use-case needs to prehash the message, it can do so. PKCS#11, for example, offers a PrehashEdDSA mechanism for cases where the message is so big that it can't be cached in the signer or the verifier. Specifically for Dilithium, this is not even necessary because H(tr || M) can be calculated by the signer/verifier without a problem by using the PKCS#11 incremental API (C_SignUpdate/ C_VerifyUpdate).

From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> **On Behalf Of** Markku-Juhani O. Saarinen
Sent: Tuesday, December 13, 2022 6:16 PM
To: pqc-forum <pqc-forum@list.nist.gov>
Cc: joost.renes <joost.renes@nxp.com>; pqc-forum <pqc-forum@list.nist.gov>; dustin...@nist.gov <dustin.moody@nist.gov>; Peter Schwabe <peter@cryptojedi.org>; Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>
Subject: RE: [EXTERNAL][EXT] [pqc-forum] Parameter selection for the selected algorithms

CAUTION: This email originated from outside of the organization. Do not click links or open attachments unless you can confirm the sender and know the content is safe.

On Tuesday, December 13, 2022 at 9:24:30 PM UTC+1 joost.renes wrote:

> Hi Markku,
>
> Would this not be solved by setting M’ = H(tr||M) and running Dilithium.Sign(M’,sk) as proposed by Peter?
>
> That would have negligible overhead since tr is computed anyway.
>
> (Though I am not convinced yet it is necessary, see questions/comments below.)

Yes, the original M'=H(tr || M) would be preferable M'=H(M). As you note, it has a negligible performance penalty, and it does provide some additional security against collision attacks.

Most modern signature schemes (EdDSA, XMSS, LMS/HSS, Falcon, Sphincs+,..) have some variant of a hash target prefix (represented here by "tr"). Prominent 90s pedigree schemes such as PCKS #1 v1.5

RSA and (EC)DSA didn't have it, and hence there is resistance in some quarters to it due to API and protocol limitations. However, I think the additional security properties are worth the migration effort.

> “Arguments can be made that the quantum security level could be less, and hence H(M) with SHA2-512 would not meet PQ Security Level 5.”
>
> I don’t really understand this point: it might have lower quantum security, but the same can be argued about AES256. The definition of the NIST security levels says SHA256 (L2) > AES128 (L1) and SHA384 (L4) > AES192 (L3). What is special about SHA512 that we do not have SHA512 > AES256 (L5)?

I see your point; an assumption has always been made that SHA256 > AES128 which sort of implies SHA512 > AES256. I agree that 512-bit collision resistance should be sufficient for Level 5 (some quantum algorithm theory person can comment in detail, but my understanding is that Grover's attack is relatively so much more effective than any PQ collision search.)

Cheers,

- markku

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

> **From:** Markku-Juhani O. Saarinen <mjos....@gmail.com>
> **Sent:** Tuesday, December 13, 2022 7:45 PM
> **To:** pqc-forum <pqc-...@list.nist.gov>
>
> **Cc:** Joost Renes <joost...@nxp.com>; pqc-forum <pqc-...@list.nist.gov>; dustin...@nist.gov <dustin...@nist.gov>; Peter Schwabe <pe...@cryptojedi.org>
> **Subject:** Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
>
> **Caution:** EXT Email
>
> On Mon, Dec 12, 2022 at 5:31 PM Joost Renes <joost...@nxp.com> wrote:
> \> We would like to put forward the option of allowing SHA-2 for the
> \> Dilithium message hash mu = H(tr || M).
>
> On Tue, Dec 13, 2022 at 3:01 PM Peter Schwabe <pe...@cryptojedi.org> wrote:
> \> Would it alternatively work for you to standardize also
> \> "HashedDilithium", i.e., a version that takes the message M, first
> \> computes M' = H(M) with any FIPS-certifiable hash function H at suitable

> > security level and then run Dilithium.Sign(M',sk)?
>
> Hi Peter and all,
>
> Allowing H(M) for Dilithium2 is potentially helpful for shortest-term transitioning pure consumer-level systems, but something different would be needed for security Level 5, which is the only version allowed in higher-security applications.
>
> Concretely: While the definition of "suitable security level" hash is reasonably clear for Dilithium2, what would it be for Dilithium5 -- especially if the "tr" prefix is removed and collision resistance is assumed?
>
> - Background note to those looking at the specs: Dilithium was already changed once during Round 3 to meet the Level 5 PQ requirement; the Round 3 submission document (v3.0 / 20201001) on the NIST website uses a weaker 384-bit "mu" length parameter; at the suggestion of external cryptanalysis this was increased to 512 bits for Dilithium v3.1 / 20210208 which is only available at the Dilithium designer's web site. https://pq-crystals.org/dilithium/resources.shtml
>
> - Removing the "tr" prefix from the initial hash opens the scheme to generic collision attacks rather than attacks targeted at a specific public key/identity. The classical security of SHA2 against these types of attacks is traditionally upper bound at the "256-bit" security level. Arguments can be made that the quantum security level could be less, and hence H(M) with SHA2-512 would not meet PQ Security Level 5.
>
> - Plain (un-prefixed) H(M) with SHA3-512 might be good enough for PQC Level 5 (SHA3 does not have a 256-bit classical attack design upper bound like SHA2), but additional quantum security analysis would be needed. Note that SHA3-512 requires 89% more permutations to process M than SHAKE256; hence it runs at almost half the speed (per kilobyte of data signed/verified) compared to the current Dilithium5 v3.1 design.
>
> - On "FIPS-certifiable:" Having had some exposure to FIPS certification, I would try to avoid this language here in particular. Even though the weakest component sets a system's security level, FIPS currently allows any combination of hashes and signature algorithms as long as all components have a classical security level of 112 bits. Typical FIPS-certified security stacks and signature/PKI solutions have a 128-bit overall classical security level, even if some individual components are designed for higher security levels ("256-bit TLS" is generally marketing language only.)
>
> Cheers,
>
> - markku

> Dr. Markku-Juhani O. Saarinen <mj...@iki.fi>
>
> On Tuesday, December 13, 2022 at 5:35:17 PM UTC+1 joost.renes wrote:
>
>> Hi Peter,
>>
>> That would alleviate the performance concerns, so yes we would be in favor
>> of having such a version in the standard.
>> Could you elaborate on why this is preferable over allowing this instance of
>> H to be a FIPS-certifiable function itself?
>> It seems to me more natural than chaining the digests as would be done in
>> hashedDilithium.
>>
>> Kind regards,
>> Joost
>>
>> -----Original Message-----
>> From: Peter Schwabe <pe...@cryptojedi.org>
>> Sent: Tuesday, December 13, 2022 3:01 PM
>> To: Joost Renes <joost...@nxp.com>
>> Cc: pqc-forum <pqc-...@list.nist.gov>; Moody, Dustin (Fed)
>> <dustin...@nist.gov>
>> Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected
>> algorithms
>>
>> Caution: EXT Email
>>
>> Joost Renes <joost...@nxp.com> wrote:
>> > Hi Dustin,
>>
>> Hi Joost,
>>
>> > We would like to put forward the option of allowing SHA-2 for the
>> > Dilithium message hash mu = H(tr || M).
>>
>> Would it alternatively work for you to standardize also "HashedDilithium",
>> i.e., a version that takes the message M, first computes M' = H(M) with any
>> FIPS-certifiable hash function H at suitable security level and then run

> > Dilithium.Sign(M',sk)?
> >
> > All the best,
> >
> > Peter

--

You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/a0917aa3-1ceb-44be-bcf4-14426cec350dn%40list.nist.gov.

--

You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/8b5c4a5467944530bf9cd6b77ceefc35%40amazon.com.

From: Peter Schwabe <peter@cryptojedi.org> via pqc-forum@list.nist.gov
To: John Mattsson <john.mattsson@ericsson.com>
CC: Kampanakis, Panos <kpanos@amazon.com>, Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>, pqc-forum <pqc-forum@list.nist.gov>, joost.renes <joost.renes@nxp.com>, dustin...@nist.gov <dustin.moody@nist.gov>, Peter Schwabe <peter@cryptojedi.org>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, December 14, 2022 03:46:21 AM ET

```
John Mattsson <john.mattsson@ericsson.com> wrote:
> Hi,

Dear all,

> I agree. I think there is a lot to learn from the EdDSA
> standardization:
>
> - Specifying PureEdDSA and PrehashEdDSA has so far just led to
> confusion. My preference would be to only have one version.
>
> - Specifying only SHA-512 in Ed25519 was likely also a bad choice.
> EdDSA has not seen any use in Web Servers. It has also not been used
> in constrained IoT where the increased performance would have been
> very welcome. Many older devices only have hardware acceleration for
> SHA-256, and many newer IoT systems is looking at using Keccak only.
> For new algorithms, I think it make sense to go for Keccak only.
```

> https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdatatracker.ietf.org%2Fmeeting%2F100%2Fmaterials%2Fslides-100-t2trg-small-crypto-for-small-iot&data=05%7C01%7Cyi-kai.liu%40nist.gov%7Ca194a61021ca4d84a65d08daddafaab6%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638066043810131026%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=RtC1Jc2%2FTrgFUn7J96L96xkbHeuBzBnrrDBQr%2Bkxn7s%3D&reserved=0<https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdatatracker.ietf.org%2Fmeeting%2F100%2Fmaterials%2Fslides-100-t2trg-small-crypto-for-small-iot-00&data=05%7C01%7Cyi-kai.liu%40nist.gov%7Ca194a61021ca4d84a65d08daddafaab6%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C638066043810131026%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=yoQXqQF%2Bv92Nf7L5q2h5TTRMIC%2FKB5Vn107eJTQxRWY%3D&reserved=0>

I very much agree with this last argument: I believe that there shouldn't be different versions of Dilithium with different hash functions. Going for Keccak only is the obvious choice and the clean solution that avoids long-term incompatibilies.

However, from Joost's comment and earlier discussions I understand that there are applications that are bottlenecked by the message hash, and currently have hardware acceleration only for SHA2. Giving these legacy applications some way to use the accelerator for the message hash would ease short-term migration.

The obvious solution is to use fast SHA2 to hash the message and then sign that hash with clean, purely Keccak-based Dilithium. I don't know exactly what is required in terms of phrasing in standards to allow this short-term solution. Somehow supporting such "pre-hashing", at least for legacy devices, seems better to me than the alternatives, namely specifying Dilithium with different options for hash functions, or not migrating those legacy devices to PQC signatures at all.

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: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, December 14, 2022 04:00:39 AM ET
Attachments: smime.p7m

Some comments seem to be requesting data regarding Ed25519 applications.
Nicolai Brown maintains a list of "Things that use Ed25519":

```
   https://ianix.com/pub/ed25519-deployment.html
```

——D. J. Bernstein

--

You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/20221214090010.32252.qmail%40cr.yp.to.

From: John Mattsson <john.mattsson@ericsson.com> via pqc-forum <pqc-forum@list.nist.gov>
To: Peter Schwabe <peter@cryptojedi.org>
CC: Kampanakis, Panos <kpanos@amazon.com>, Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>, pqc-forum <pqc-forum@list.nist.gov>, joost.renes <joost.renes@nxp.com>, dustin...@nist.gov <dustin.moody@nist.gov>, Peter Schwabe <peter@cryptojedi.org>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, December 14, 2022 04:24:31 AM ET

Peter Schwabe wrote:

> there are applications that are bottlenecked by the message hash, andcurrently

> have hardware acceleration only for SHA2.

My experience from working a lot with constrained device developers is that that a lot of current devices have acceleration of SHA-256 only, i.e., not SHA2 in general.

John

From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> on behalf of Peter Schwabe <peter@cryptojedi.org>
Date: Wednesday, 14 December 2022 at 09:46
To: John Mattsson <john.mattsson@ericsson.com>
Cc: Kampanakis, Panos <kpanos@amazon.com>, Markku-Juhani O. Saarinen <mjos.crypto@gmail.com>, pqc-forum <pqc-forum@list.nist.gov>, joost.renes <joost.renes@nxp.com>, dustin...@nist.gov <dustin.moody@nist.gov>, Peter Schwabe <peter@cryptojedi.org>
Subject: Re: [EXT] [pqc-forum] Parameter selection for the selected algorithms

John Mattsson <john.mattsson@ericsson.com> wrote:

> Hi,

Dear all,

> I agree. I think there is a lot to learn from the EdDSA
> standardization:
>
> - Specifying PureEdDSA and PrehashEdDSA has so far just led to

> confusion. My preference would be to only have one version.
>
> - Specifying only SHA-512 in Ed25519 was likely also a bad choice.
> EdDSA has not seen any use in Web Servers. It has also not been used
> in constrained IoT where the increased performance would have been
> very welcome. Many older devices only have hardware acceleration for
> SHA-256, and many newer IoT systems is looking at using Keccak only.
> For new algorithms, I think it make sense to go for Keccak only.
> https://datatracker.ietf.org/meeting/100/materials/slides-100-t2trg-small-crypto-for-small-iot<https://datatracker.ietf.org/meeting/100/materials/slides-100-t2trg-small-crypto-for-small-iot-00>

I very much agree with this last argument: I believe that there
shouldn't be different versions of Dilithium with different hash
functions. Going for Keccak only is the obvious choice and the clean
solution that avoids long-term incompatibilies.

However, from Joost's comment and earlier discussions I understand that
there are applications that are bottlenecked by the message hash, and
currently have hardware acceleration only for SHA2. Giving these
legacy applications some way to use the accelerator for the message hash
would ease short-term migration.

The obvious solution is to use fast SHA2 to hash the message and then
sign that hash with clean, purely Keccak-based Dilithium. I don't know
exactly what is required in terms of phrasing in standards to allow this
short-term solution. Somehow supporting such "pre-hashing", at least for
legacy devices, seems better to me than the alternatives, namely
specifying Dilithium with different options for hash functions, or not
migrating those legacy devices to PQC signatures at all.

All the best,

Peter

--

forum+unsubscribe@list.nist.gov.

To view this discussion on the web visit https://protect2.fireeye.com/v1/url?k=31323334-501d5122-313273af-454445555731-22b59aabc9945a1d&q=1&e=634b4a88-7c3a-445e-a650-263e4aa23a24&u=https%3A%2F%2Fgroups.google.com%2Fa%2Flist.nist.gov%2Fd%2Fmsgid%2Fpqc-forum%2FY5mNNklNrkGJqtwt%2540disp3269.

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Friday, January 20, 2023 01:57:57 PM ET
Attachments: smime.p7m

NIST claimed in its round-3 report in July 2022 that Kyber-512 qualifies for "category 1", i.e., is as hard to break as AES-128, the minimum security level allowed in NISTPQC.

("Figure 1 shows [performance] for Kyber ... for security categories 1 and 3"; Figure 1 lists Kyber-512 and Kyber-768.)

When NIST started this thread in November 2022, it announced plans to standardize Kyber-512, and claimed that Kyber-512 is "likely" as hard to break as AES-128 in "in realistic models of security" that account for the costs of memory, assuming no "major improvements in cryptanalysis".

NIST's email dated 7 Dec 2022 22:38:45 +0000 _sounds_ like it includes the components of NIST's calculations of security margins (or deficits) for Kyber-512 in a particular space of scenarios. (NIST says that the scenarios with deficits are unlikely.)

However, NIST didn't give any clear end-to-end statements that Kyber-512 has N bits of security margin in scenario X for clearly specified (N,X).

In the absence of such clarity, reviewers have to worry that putting NIST's stated components together in what _seems_ to be the obvious way, and then doing the work to disprove what NIST _appears_ to be claiming about the security margin, will lead to a response claiming that, no, NIST meant something else. It's natural to ask for clarification.

So I picked a simple scenario that's extremely favorable to Kyber-512, and asked (see quote below) whether I was correctly gathering that NIST was claiming 34 bits of security margin for Kyber-512 in that scenario. I went through exactly which NIST scenario I was picking and what NIST seemed to be calculating in that scenario.

I was hoping for a prompt "Yes, that's correct" answer. But NIST still hasn't replied.

I've again gone through NIST's 7 December email, and again concluded that for this scenario NIST is claiming 34 bits in the way spelled out below. Is there any way I could be missing something here? Does anyone see another way to interpret NIST's calculations?

——D. J. Bernstein

```
> Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
> From: "D. J. Bernstein" <djb@cr.yp.to>
> To: pqc-forum@list.nist.gov
> Message-ID: <20221208021006.507522.qmail@cr.yp.to>
>
> 'Perlner, Ray A. (Fed)' via pqc-forum writes:
> > We can elaborate a little bit further on our reasoning leading to our
> > current assessment that Kyber512 likely meets NIST category I
>
> Thanks. I've read through, and I have a much more specific clarification
> question to make sure I understand the underlying calculations. Within
> the space of scenarios reviewed, if we take the particular scenario of
>
>    (1) assuming accuracy of 2^137 from the most recent attack paper
>        taken into account (Matzov) regarding the number of "gates",
>
>    (2) assuming this isn't affected by the "known unknowns", and
>
>    (3) assuming accuracy of the RAM-cost model in the NTRU Prime
>        documentation,
>
> then am I correctly gathering that you're calculating the Kyber-512
> security level as 2^177 (i.e., 34 bits of security margin compared to
> 2^143 for AES-128), where this 177 comes from the above 137 plus 40,
> where 40 comes from 169 minus 129 on page 103 of the NTRU Prime
```

> documentation, specifically "real" minus "free" for pre-quantum sieving
> for sntrup653?

--

From: David A. Cooper <david.cooper@nist.gov> via pqc-forum <pqc-forum@list.nist.gov>
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Monday, January 23, 2023 06:07:42 PM ET

On 1/20/23 10:57 AM, D. J. Bernstein wrote:

> NIST claimed in its round-3 report in July 2022 that Kyber-512 qualifies for "category 1", i.e., is as hard to break as AES-128, the minimum security level allowed in NISTPQC. ("Figure 1 shows [performance] for Kyber ... for security categories 1 and 3"; Figure 1 lists Kyber-512 and Kyber-768.)

As the primary author of Section 2.2.2 of NISTIR 8413, I would like to clarify that everything written in that section was based on the submitters' claimed security categories of each of the parameter sets for each of the schemes. It was not my intention (nor of anyone who helped in writing that section) to make any assertions about whether or not the parameter sets actually provided the claimed level of security. I apologize if my failure to note this in the text caused any confusion.

> When NIST started this thread in November 2022, it announced plans to standardize Kyber-512, and claimed that Kyber-512 is "likely" as hard to break as AES-128 in "in realistic models of security" that account for the costs of memory, assuming no "major improvements in cryptanalysis". NIST's email dated 7 Dec 2022 22:38:45 +0000 _sounds_ like it includes the components of NIST's calculations of security margins (or deficits) for Kyber-512 in a particular space of scenarios. (NIST says that the scenarios with deficits are unlikely.) However, NIST didn't give any clear end-to-end statements that Kyber-512 has N bits of security margin in scenario X for clearly specified (N,X). In the absence of such clarity, reviewers have to worry that putting NIST's stated components together in what _seems_ to be the obvious way, and then doing the work to disprove what NIST _appears_ to be claiming about the security margin, will lead to a response claiming that, no, NIST meant something else. It's natural to ask for clarification.

The email you cited (https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/4MBurXr58Rs/m/xHojUDCaBAAJ), speaks for itself. NIST continues to be interested in people's opinions on whether or not our current plan to standardize Kyber512 is a good one. While reviewers are free, as a fun exercise, to attempt to "disprove what NIST _appears_ to be claiming about the security margin," the results of this exercise would not be particularly useful to the standardization process.NIST's prior assertions and their interpretation are not relevant to the question of whether people believe that it is a good idea to standardize Kyber512.

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, January 24, 2023 10:40:00 AM ET
Attachments: smime.p7m

'David A. Cooper' via pqc-forum writes:

> The email you cited (https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/4MBurXr58Rs/m/xHojUDCaBAAJ),
> speaks for itself.

I _think_ I understand what NIST is claiming in that message regarding the quantitative Kyber security level.

I _think_ that my clarification question (focusing on one example, much shorter than NIST's message) is identifying the obvious interpretation.

But then why hasn't NIST simply said "Yes, that's correct" in response?

If the interpretation I've identified differs from what NIST meant, can NIST please simply say what the difference is, so that security reviewers don't have to spend time on the quantitative security claims that NIST currently _seems_ to be making?

> the results of this exercise would not be particularly useful to the
> standardization process.

Um, what?

If Kyber-512 doesn't meet the minimum security level allowed by the official call for submissions to the NIST Post-Quantum Cryptography Standardization Project then Kyber-512 should not be standardized.

NIST's evaluation of the Kyber-512 security level—after various attack advances newer than the latest version of the Kyber submission—depends explicitly on NIST's calculations of the impact of memory costs.

With all due respect, is it so hard to imagine that NIST has botched those calculations? If NIST is so sure that it got the whole sequence of calculations right, why is it so resistant to clarification questions that will help reviewers check and confirm that NIST got this right? If NIST _isn't_ sure, doesn't that make public review even more important?

In any case, there's a strong public interest in having NIST's security evaluations clearly and promptly explained, to maximize the chance of having errors corrected before bad decisions are set into stone.

> As the primary author of Section 2.2.2 of NISTIR 8413, I would like to clarify that everything written in that section was based on the submitters' _claimed_ security categories of each of the parameter sets for each of the schemes.

Hmmm. When NIST wrote "Figure 1 shows [performance] for Kyber ... for security categories 1 and 3", NIST was explicitly labeling those parameter sets as having categories 1 and 3.

To be clear, you're saying the intent was merely to report what _the submission claimed for the security levels_, not to have _NIST_ on record expressing this security evaluation?

If so, an erratum and updated report are warranted. Not being careful about distinguishing the statements "X" and "Y said X" makes it much too easy to slide without justification from one into the other.

It's clear either way that NIST's more recent announcements of plans to standardize Kyber-512, and more recent claims regarding the Kyber-512 security level, are going beyond what the report said. But the report was already informing the reader of a central NIST conclusion that Kyber-512 meets category 1. People often take timelines into account in evaluating risks (for example, earlier conclusions obviously can't have accounted for more recent attacks), so if the conclusion was actually reached later then this should be made clear.

---D. J. Bernstein

From: Christopher J Peikert <cpeikert@alum.mit.edu> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, January 24, 2023 12:44:10 PM ET

On Tue, Jan 24, 2023 at 10:39 AM D. J. Bernstein <djb@cr.yp.to> wrote:

> 'David A. Cooper' via pqc-forum writes:
> \> The email you cited (https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/4MBurXr58Rs/m/xHojUDCaBAAJ),
> \> speaks for itself.
>
> I _think_ that my clarification question (focusing on one example, much shorter than NIST's message) is identifying the obvious interpretation.

Having just re-read both messages, I don't think that's the obvious interpretation of NIST's message -- or even a plausible one.

Your clarifying question asked whether NIST was "calculating the Kyber-512 security level as 2^177 (i.e., 34 bits of security margin compared to 2^143 for AES-128), where this 177 comes from the above 137 plus 40, where 40 comes from 169 minus 129 on page 103 of the NTRU Prime documentation..."

What NIST actually said (key points excerpted, emphasis added):

- "...assessments by the NTRU and NTRUprime teams gave estimates that would suggest the cost of sieving against category 1 parameters, in models that account for the cost of memory access, is **something like 20 to 40 bits of security more** than would be suggested by the RAM model..."
- "Taking Matzov's estimates of the attack cost to be accurate, only 6 bits of security from memory access costs are required for Kyber512 to meet category 1, so **in this case Kyber512 would meet category 1 even if the NTRU and NTRUprime submission significantly overestimate the cost of memory access** in lattice sieving algorithms. Further, ..."

The obvious interpretation is that NIST is **not** claiming as much as 40 - 6 = 34 bits of security margin (compared to 2^143 for AES-128).

Sincerely your in cryptography,

Chris

--

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, January 24, 2023 03:59:05 PM ET
Attachments: smime.p7m

NIST _seems_ to be saying that scenario X gives 2^177, scenario Y gives 2^157, etc. This is structured in a way that puts a lot of work on the reader: each individual analysis step is mixed with scenario-likelihood evaluations, and there are zero examples of confirming tallies.

After doing this work, I named specifically X as an example, gave a much simpler review of how NIST _seems_ to be calculating 177 for scenario X, and asked NIST for confirmation that I was understanding NIST's message correctly. I was expecting a prompt "Yes, that's correct" answer.

Comments along the lines of "You say that NIST is claiming X and 177, but that's not a plausible interpretation, since obviously they're considering Y and 157 as a possibility too!" are attacking a strawman.

Christopher J Peikert writes:
> Your clarifying question asked whether NIST was "calculating the Kyber-512
> security level as 2^177 (i.e., 34 bits of security margin compared to 2^143
> for AES-128), where this 177 comes from the above 137 plus 40, where 40
> comes from 169 minus 129 on page 103 of the NTRU Prime documentation... "

No, your quote omits essential context: "Within the space of scenarios reviewed, if we take the particular scenario of (1) assuming accuracy of 2^137 from the most recent attack paper taken into account (Matzov) regarding the number of 'gates', (2) assuming this isn't affected by the 'known unknowns', and (3) assuming accuracy of the RAM-cost model in the NTRU Prime documentation, then am I correctly gathering ... "

> What NIST actually said (key points excerpted, emphasis added):

Here you highlight some of their text that's obviously considering not just X, but also scenarios less favorable to Kyber. The combination of

(1) this highlighting,
(2) the omission of essential context, and
(3) the claim that my interpretation is implausible

makes it sound as if the request was for confirmation that NIST is saying _X and 177_. (Obviously their message didn't say that.)

Anyone who reads my full message sees that this is not even close to a correct characterization of my question. I began by saying that this was just one of the scenarios reviewed. I'm simply asking for confirmation that NIST is claiming security level 2^177 _if_ we're in this scenario, where the 177 is calculated in the way I reviewed.

---D. J. Bernstein

From: Perlner, Ray A. (Fed) <ray.perlner@nist.gov> via pqc-forum <pqc-forum@list.nist.gov>
To: D. J. Bernstein <djb@cr.yp.to>, pqc-forum <pqc-forum@list.nist.gov>
Subject: RE: [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, January 24, 2023 05:53:49 PM ET

Dan and Chris,

It would be helpful to redirect discussion to

1) The question of whether Kyber512 is as hard to break as AES128, (which is a scientific question that cannot be settled by NIST pronouncements)
2) The related question of whether Kyber512 should be standardized, (which is a question where NIST will ultimately need to make a definitive decision, but thus far we have only signaled we are leaning towards yes.)

With this in mind, I would like to note that the technical point on which Dan has asked for clarification is effectively "how much additional security does Kyber512 get on account of memory access costs, according to the NTRUprime submission's memory cost model?" Surely Dan, being on the NTRUPrime team, is in a better position to answer this question than us.

Ray

-----Original Message-----
From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> On Behalf Of D. J. Bernstein
Sent: Tuesday, January 24, 2023 3:59 PM
To: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms

NIST _seems_ to be saying that scenario X gives 2^177, scenario Y gives 2^157, etc. This is structured in a way that puts a lot of work on the reader: each individual analysis step is mixed with scenario-likelihood evaluations, and there are zero examples of confirming tallies.

After doing this work, I named specifically X as an example, gave a much simpler review of how NIST _seems_ to be calculating 177 for scenario X, and asked NIST for confirmation that I was understanding NIST's message correctly. I was expecting a prompt "Yes, that's correct" answer.

Comments along the lines of "You say that NIST is claiming X and 177, but that's not a plausible interpretation, since obviously they're considering Y and 157 as a possibility too!" are attacking a strawman.

Christopher J Peikert writes:
> Your clarifying question asked whether NIST was "calculating the
> Kyber-512 security level as 2^177 (i.e., 34 bits of security margin
> compared to 2^143 for AES-128), where this 177 comes from the above
> 137 plus 40, where 40 comes from 169 minus 129 on page 103 of the NTRU Prime
documentation ... "

No, your quote omits essential context: "Within the space of scenarios reviewed, if we take the particular scenario of (1) assuming accuracy of
2^137 from the most recent attack paper taken into account (Matzov) regarding the number of 'gates', (2) assuming this isn't affected by the 'known unknowns', and (3) assuming accuracy of the RAM-cost model in the NTRU Prime documentation, then am I correctly gathering ... "

> What NIST actually said (key points excerpted, emphasis added):

Here you highlight some of their text that's obviously considering not just X, but also scenarios less favorable to Kyber. The combination of

(1) this highlighting,
(2) the omission of essential context, and
(3) the claim that my interpretation is implausible

makes it sound as if the request was for confirmation that NIST is saying _X and 177_. (Obviously their message didn't say that.)

Anyone who reads my full message sees that this is not even close to a correct characterization of my question. I began by saying that this was just one of the scenarios reviewed. I'm simply asking for confirmation that NIST is claiming security level 2^177 _if_ we're in this scenario, where the 177 is calculated in the way I reviewed.

———D. J. Bernstein

--

--

From: Christopher J Peikert <cpeikert@alum.mit.edu> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, January 24, 2023 06:00:14 PM ET

On Tue, Jan 24, 2023 at 3:59 PM D. J. Bernstein <djb@cr.yp.to> wrote:

> NIST _seems_ to be saying that scenario X gives 2^177, scenario Y gives
> 2^157, etc.

Again, I don't think this is "the obvious interpretation" of what NIST wrote.

In particular, I don't read any claim of N-6 bits of security over AES-128, for N=40 or any other value, in any specific scenario X, Y, Z, or otherwise.

Sincerely yours in cryptography,

Chris

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] Parameter selection for the selected algorithms
Date: Tuesday, January 24, 2023 10:12:27 PM ET
Attachments: smime.p7m

This message is going step by step through NIST's 7 Dec 2022 22:38:45 +0000 posting, in particular tracing through exactly how that posting _appears_ to be quantifying the Kyber-512 security level—a topic of obvious importance, given NIST's plans to standardize Kyber-512.

If anyone sees any way that I could be misunderstanding the details of NIST's posting, please pinpoint which step is at issue and what the alternative interpretation of NIST's calculation is supposed to be.

As an initial comment, one of the complications in NIST's posting is that it considers a large space of scenarios, with analysis steps mixed into comments on the likelihood of each scenario. NIST doesn't give any confirming end-to-end examples of the tallies obtained by putting the steps together in what _seems_ to be the obvious way.

In early December, I picked one scenario—let's call it scenario X—from within the space that NIST's posting had specified. Scenario X makes the following three assumptions:

(1) Assume accuracy of 2^137 from the most recent attack paper taken into account (Matzov) regarding the number of "gates". (This is a number specifically mentioned by NIST. NIST also considers the more complicated possibility of this estimate being invalid.)

(2) Assume this isn't affected by the "known unknowns". (This is a possibility specifically mentioned by NIST. NIST also considers the more complicated possibility of the security level being affected by the "known unknowns".)

(3) Assume accuracy of the RAM-cost model in the NTRU Prime documentation. (This is one of two sources that NIST repeatedly

points to and calculates numbers on the basis of. NIST also considers other possibilities for the RAM cost.)

Obviously NIST's quantitative conclusions vary depending on the exact assumptions. The reason I asked specifically about scenario X is that this is simpler than considering the full space of scenarios. I'll continue using scenario X as an illustrative example in this message, along with giving examples of how it simplifies the analysis.

Back in December, I stated the full definition of X, stated my understanding of exactly how NIST was calculating the Kyber-512 security level in particular for scenario X, and asked for confirmation.

I was expecting prompt confirmation. So far NIST has neither confirmed nor denied. I'm baffled by the explanations I've seen for this lack of clarification.

> We can elaborate a little bit further on our reasoning leading to our
> current assessment that Kyber512 likely meets NIST category I (similar
> considerations apply to the other parameter sets we plan to
> standardize for lattice-based schemes.)

This is a preliminary statement regarding the importance of the calculations at hand. See below for the calculations.

> That said, beyond this message, we don’t think further elaboration of
> our current position will be helpful. While we did consult among
> ourselves and with the Kyber team,

I filed a formal complaint in December regarding NIST's lack of transparency regarding its investigation of Kyber-512 security. I filed a new FOIA request in mid-January.

> it’s basically just our considered
> opinion based on the same publicly available information everyone else
> has access to.

No. NIST's posting starts from, e.g., the Matzov paper's 2^137 estimate for "gates", but then goes beyond this in quantifying the impact of memory costs, something the Matzov paper definitely did not do.

Scenario X explicitly assumes accuracy of the Matzov paper's 2^137 estimate for "gates". I'm not asking NIST to explain where that number comes from; I'm asking for confirmation of my understanding of what NIST is calculating _starting_ from that number.

> The point of this thread is to seek a broader range of perspectives on whether our current plan to standardize Kyber512 is a good one, and a long back and forth between us and a single researcher does not serve that purpose.

Public review of NIST's security evaluations requires transparency and clarity regarding those evaluations. It is not appropriate for NIST to be asking for a range of perspectives while concealing information. An open and transparent process would involve less "back and forth" than the process that NIST has chosen.

> Here's how we see the situation:
> In April this year, “Report on the Security of LWE” was published by MATZOV (https://zenodo.org/record/6412487#.Y4-V53bMKUk), describing an attack, assessed in the RAM model to bring some parameter sets, including Kyber512, slightly below their claimed security strength categories.

This is the most recent attack paper mentioned in NIST's posting. That's why my description of scenario X says "the most recent attack paper taken into account (Matzov)".

It's surprising that NIST's posting doesn't mention any of the newer attack papers. Hypothesizing that there are no "major improvements in cryptanalysis" doesn't justify ignoring the improvements that have already been published!

Anyway, given that NIST is calculating security levels starting from the

Matzov paper, I'd like to make sure I understand those calculations.

"Assessed in the RAM model" appears to be referring to the Matzov paper counting the number of "gates". As a side note, "the" RAM model is ambiguous; the literature defines many different models (and many different sets of "gates").

> In particular, the report estimates the cost of attacking Kyber512
> using a classical lattice attack to be 2^137 bit operations, which is
> less than the approximately 2^143 bit operations required to
> classically attack AES-128.

NIST takes this 137 as the foundation of various calculations below.

This doesn't mean NIST is saying Kyber-512 is broken in 2^137 "gates". NIST is saying that Matzov estimated 137, and then NIST is calculating various consequences of the 137. If the 137 is inaccurate then the details of NIST's calculations (see below) would go up or down accordingly.

For purposes of putting together the sources available, the simplest case to consider is that 2^137 accurately counts the number of "gates". Scenario X explicitly assumes this.

> However, like previous lattice attacks, the MATZOV attack is based on
> sieving techniques, which require a large amount of (apparently
> unstructured) access to a very large memory.

The whole starting point here is NIST's 30 Nov 2022 12:25:47 +0000 announcement of its plans to standardize Kyber-512.

As justification, this announcement claims that "the best known attacks against Kyber-512 require huge amounts of memory and the real attack cost will need to take the cost of (access to) memory into account. ... barring major improvements in cryptanalysis, it seems unlikely that the cost of memory access will ever become small enough to cause Kyber-512 to fall below category 1 security, in realistic models of security that take these costs into account".

The posting I'm currently commenting on is repeating this conclusion, and then filling in some supporting calculations. See below.

> The RAM model ignores the cost of this memory access,

Indeed, the "gate" counts in question ignore memory access.

> and while the science of comparing the cost of memory access to other costs involved in a large cryptanalytic attack is not as mature as we would like, it seems overwhelmingly likely that, in any realistic accounting of memory access costs, these will significantly exceed the costs that are assessed by the RAM model for lattice sieving.

There are three obvious questions at this point.

First, what exactly does "significantly" mean in this context?

Second, how does NIST reach its "overwhelmingly likely ... significantly exceed" conclusion?

Third, how does NIST get from "significantly exceed" to its conclusion that having Kyber-512 fall short of AES-128 is "unlikely"? (Assuming no "major improvements in cryptanalysis".)

> The largest practical implementation of sieving techniques we know of, described in detail in “Advanced Lattice Sieving on GPUs, with Tensor Cores” by Ducas, Stevens, and van Woerden (https://eprint.iacr.org/2021/141), was forced by memory access limitations, to adopt settings for bucket size, that would be suboptimal according to the RAM model.

Is "bucket size ... suboptimal" supposed to imply NIST's "significantly" claim regarding "costs", and, from there, NIST's claim that it's "unlikely" for Kyber-512 to be easier to break than AES-128?

There's still no quantification by this point in NIST's posting. This makes review practically impossible. However, NIST does present quantified calculations later.

> It should be noted that, increasing the scale of the instances being attacked to near cryptographic scale would probably require extensive hardware optimization, e.g. by using special purpose ASICs, and these techniques, being generally acknowledged to be less effective against memory-intensive tasks, would likely make memory access even more of a bottleneck.

Qualitatively, this is a reasonable summary of what the literature on point is saying, but how does NIST get from this to the claim that Kyber-512 is "unlikely" to be below the AES-128 security level?

> Additionally,

This is where NIST's posting transitions into quantification.

> While the Kyber, Dilithium, and Falcon teams did not give a quantitative assessment of the practical cost of memory access during sieving against cryptographic parameters, assessments by the NTRU and NTRUprime teams gave estimates that would suggest the cost of sieving against category 1 parameters, in models that account for the cost of memory access, is something like 20 to 40 bits of security more than would be suggested by the RAM model.

Finally some numbers to work with! See below for how NIST uses these numbers.

As a side note, NIST seems to have very low confidence in these numbers, saying not just "estimates" but also "suggest" and "something like". But my question is _not_ about confidence levels in the estimates.

What I want to make sure I understand is simply NIST's calculations _starting from_ these estimates: how NIST is drawing conclusions about Kyber-512 _if_ it hypothesizes, e.g., 40.

That's why scenario X explicitly assumes accuracy of one of the two sources that NIST cited. I picked NTRU Prime; in context, this choice of source is favorable to Kyber.

> (For NTRU’s estimates see section 6.3 of the round 3 specification document available at https://ntru.org/index.shtml . For NTRUprime’s estimates see section 6.11 of https://ntruprime.cr.yp.to/nist/ntruprime-20201007.pdf .

Scenario X is specifically assuming "accuracy of the RAM-cost model in the NTRU Prime documentation", one of the two sources that NIST relies upon for its quantification. See below for the numbers that NIST obtains from this source.

> The Kyber spec (available at https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf) discusses, but does not quantify, memory access costs in section 5.3 (Q6))

Indeed, what's cited here doesn't quantify this. So let's keep going with the numbers that NIST obtained from other sources.

> Taking Matzov's estimates of the attack cost to be accurate,

This is exactly what scenario X is assuming. Of course, NIST's posting also considers other possibilities, but let's follow through what NIST obtains from this assumption.

> only 6 bits of security from memory access costs are required for Kyber512 to meet category 1,

Indeed, 137 is "only" 6 bits short of the 143 goal. NIST wants to find 6 bits of security that it can credit to Kyber-512, and says that the costs of memory do the job.

> so in this case Kyber512 would meet category 1 even if the NTRU and NTRUprime submission significantly overestimate the cost of memory

> access in lattice sieving algorithms.

Here NIST is finding more than its desired 6 bits of security: namely, the "20 to 40 bits" coming from "assessments by the NTRU and NTRUprime teams" of the extra costs coming from memory access.

For example, if NTRU says 20 and if this is accurate, then NIST is calculating a security level of 137+20 = 157, safely above 143. (Again, this is explicitly assuming accuracy of the 137 in the first place.)

This seems really straightforward. I can't see how NIST's text can be interpreted in any other way. Does anyone see how I could be missing something here?

As another example, if NTRU Prime says 40 and if this is accurate, then NIST is calculating a security level of 137+40 = 177, even farther above 143. (Once again assuming accuracy of the 137.)

NIST says that even if those sources have "significantly" overestimated the memory-access cost then Kyber-512 is still okay. Working backwards from NIST's desired conclusion suggests that the "significant" boundary here is set as 14 bits compared to NTRU. Scenario X skips this question by simply assuming accuracy of the NTRU Prime RAM-cost model.

> Further, since about 5 of the 14 claimed bits of security by Matzov
> involved speedups to local computations in AllPairSearch (as described
> by section 6 of the MATZOV paper), it is likely that Kyber512 would
> not be brought below category 1 by the MATZOV attack, as long as
> state of the art lattice cryptanalyses prior to the MATZOV paper were
> bottlenecked by memory at all.

It's of course correct that if there's a bottleneck then speeding up computations outside the bottleneck has little impact. See below for how NIST seems to be using this.

> However, we acknowledge there is some additional uncertainty in the
> exact complexity of the MATZOV attack (and all other sieving-based

> lattice attacks) due to the known-unknowns Dan alludes to (described
> with quantitative estimates in section 5.3 of the Kyber spec.)

Three reasons that it might be possible to beat Matzov's 2^137 "gates" are (1) inaccuracies in Matzov's analysis (of course, these could also point the other way), (2) missing optimizations covered by the "known unknowns", and (3) missing optimizations beyond the "known unknowns".

Here NIST is pointing to #2. As a side note, it's disturbing to not see NIST accounting for #1 and #3. NIST explicitly assumed that there are no "major" improvements in cryptanalysis, but some of its scenarios have Kyber with very few bits of security margin, and closing those wouldn't require "major" improvements.

Scenario X skips this complication: it explicitly assumes that the 137 is accurate, and that there are no improvements from the "known unknowns".

> Nonetheless, even taking the most paranoid values for these
> known-unknowns (16 bits of security loss),

This is what the Kyber documentation says is the worst case, yes.

> the cost of memory access and/or algorithmically making memory access
> local, would still need to be less than what both the NTRU and
> NTRUPrime submissions assume.

I found this puzzling when I first read it: if we take 137, and then subtract a hypothesized 16, then we need to make up 22 bits, which is less than the 40 that NIST mentioned but _not_ less than the 20. What's going on?

The best explanation I could come up with is that NIST thinks the 16 overlap the 5 bits that NIST mentioned above from Matzov, so NIST is actually taking 137-16+5, meaning that NIST has to find only 17 bits, and then the 20 that NIST attributes to NTRU is enough (at least if we disregard the uncertainties conveyed by "estimate" and "suggest" and "something like").

Again, Scenario X simply assumes that the 137 is accurate, with no speedups from the "known unknowns", so this complication doesn't arise for that scenario.

> The low end estimate of approximately 20 bits (from the NTRU submission) is based on a conjecture by Ducas that a fully local implementation of the BGJ1 sieving algorithm is possible.

Here NIST is pointing to a reason to ask whether the NTRU model is too low. Scenario X explicitly takes the NTRU Prime model, which doesn't trigger this particular issue.

> So, in the case that all known-unknowns take on the most paranoid values, this would either require a sieving algorithm with local memory access that is much better than any such published algorithm, and in fact better than any that has been conjectured (at least as far as we are aware),

This is summarizing NIST's calculations from the perspective of what algorithmic improvements would be required to break NIST's conclusions. This isn't relevant to scenario X.

> or it would require the approximately 40 bits of additional security quoted as the "real cost of memory access" by the NTRUprime submission to be a massive overestimate.

This is summarizing NIST's calculations from the perspective of what modeling errors would be required to break NIST's conclusions.

I'm concerned about deviations between what NIST attributes to its source here and what the source actually says. For example, the source says that it's _estimating_ the cost of memory access, whereas NIST incorrectly makes it sound as if an estimate is being mislabeled as a fact. More to the point, I don't see how the 40 bits that NIST claims as memory overhead is a quote from what the source says on this topic.

I presume that NIST obtained 40 in the following easy way: look at the security-level table on page 103 of the source; observe that pre-quantum sieving for sntrup653 at the top is listed as 169 and 129 for "real" and "free" respectively; subtract the 129 from the 169.

If I'm correct in thinking that NIST came up with the 40 in this way, then NIST should simply say so. If NIST actually came up with the 40 in another way, then NIST should pinpoint what exactly it's attributing to the source that it's citing and how exactly NIST obtained the 40 from that.

Putting everything together, my request to NIST is simply to confirm (1) how NIST is obtaining the 40 and (2) that it's adding the 40 to the 137 to obtain a 177 security level for Kyber-512 _if_ we assume scenario X. This is much simpler than going through the full space of scenarios.

> In any event, a lot of things would have to go wrong simultaneously to push the real-world classical cost of known attacks against Kyber512 below category 1, which is why we don't think it's terribly likely.

This is going beyond the per-scenario calculations into an assessment of the probability of each scenario.

> As a final note, known quantum speedups for lattice sieving are much less effective than Grover’s algorithm for brute force key search, so in the likely scenario where the limiting attack on AES128 is Grover’s algorithm, this would further increase the security margin of Kyber512 over AES128 in practice.

This is yet another complication, and one with several unquantified steps. I'm merely asking for confirmation of my understanding of NIST's calculations regarding Kyber-512's _pre-quantum_ security level.

---D. J. Bernstein

--

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Tuesday, January 24, 2023 11:48:01 PM ET
Attachments: smime.p7m

'Perlner, Ray A. (Fed)' via pqc-forum writes:

> With this in mind, I would like to note that the technical point on
> which Dan has asked for clarification is effectively "how much
> additional security does Kyber512 get on account of memory access
> costs, according to the NTRUprime submission's memory cost model?"

NIST cited NTRU and NTRU Prime as sources estimating that, in NIST's words, the costs of memory add "something like 20 to 40 bits of security more than would be suggested by the RAM model".

NIST considered "Taking Matzov's estimates of the attack cost to be accurate" as a scenario. In this scenario, Matzov's 137 for Kyber-512, plus "20 to 40 bits", gives 157 to 177, leaving a margin above 143.

Equivalently, in this scenario, what needs to be added to the 137 to reach 143 is "only 6 bits of security from memory access costs", in NIST's words, so "20 to 40 bits of security more" than 137 are more than enough.

I've gone very carefully through NIST's message. I don't see how the "20 to 40 bits of security more" can be understood in any other way, nor do I see where anyone has presented an alternative interpretation. All I'm asking for is

* confirmation that, _if_ we simply assume accuracy of the 40 and of Matzov's 2^137 "gates", _then_ NIST is calculating the Kyber-512 security level as "40 bits of security more", meaning 2^177; and

* a clear statement for the record of how NIST obtained the 40 that NIST attributes to the NTRU Prime documentation (I presume from the 169 and 129 stated for sntrup653).

I'm not asking NIST for new analysis. NIST already posted calculations as part of justifying its announced plan to standardize Kyber-512, and I simply want to make sure I understand those calculations.

---D. J. Bernstein

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] Parameter selection for the selected algorithms
Date: Wednesday, January 25, 2023 10:44:16 AM ET
Attachments: smime.p7m

Christopher J Peikert writes:

> In particular, I don't read any claim of N-6 bits of security over AES-128,
> for N=40 or any other value, in any specific scenario X, Y, Z, or otherwise.

Are you saying that NIST's "Taking Matzov's estimates of the attack cost to be accurate" isn't considering the scenario that attacks cost 2^137 in what NIST calls "the RAM model"?

Are you saying that NIST's "something like 20 to 40 bits of security more than would be suggested by the RAM model" isn't considering the scenarios of adding, e.g., 20 or 40 to the above 137?

Are you saying that 137+20 isn't 157? Or that 137+40 isn't 177? NIST's "20 to 40 bits ... more than" doesn't mean normal mathematical addition?

I'm not asking NIST to endorse the accuracy of the sources it's using. I'm simply asking NIST to confirm

(1) that, when it started from "Matzov's estimate" and then wrote "40 bits of security more than what would be suggested by the RAM model", NIST was referring to security level 2^177; and

(2) that NIST obtained this 40 from 169 minus 129 on page 103 of the NTRU Prime documentation, specifically "real" minus "free" for pre-quantum sieving for sntrup653.

Regarding #2, NIST says "the approximately 40 bits of additional security quoted as the 'real cost of memory access' by the NTRUprime submission"—but I don't see the source stating this 40. NIST should explain how NIST obtained the 40 that it attributes to this source.

I came up with a simple explanation, namely the above 169-129, but so far NIST hasn't confirmed that this is how it obtained the 40. It is not appropriate for NIST to be using _my_ position on the NTRU Prime team as an excuse to avoid clearly answering the question of how _NIST_ obtained this 40 from the NTRU Prime documentation.

Regarding #1, I don't see how the original text leaves any ambiguities. If NIST's text allows a different interpretation, why has nobody been able to say what that alternative interpretation is? If NIST meant something else, why hasn't it stated for the record what it meant?

It's weird that asking for confirmation of what _seems_ to be the plain meaning of NIST's "more than" text---also matching the surrounding text in NIST's posting; I've gone through every word of this in detail---is producing so much resistance. Examples of the responses so far:

* NIST's posting "speaks for itself". (Um, why not confirm that my understanding is correct, or else say what I missed?)

* NIST was considering not just the 137 and the 40 but also further possibilities. (Yes, obviously, which is why my question began by explicitly focusing on one simple scenario as an example.)

* I'm asking NIST for new technical evaluation. (No, I'm asking NIST a clarification question about calculations that NIST posted.)

* Discussions of NIST's evaluation of the Kyber-512 security level should be terminated. (Um, what? This NIST security evaluation is the foundation of NIST's announced plan to standardize Kyber-512!)

Et cetera. None of this is proposing an alternative interpretation of NIST's text, and all of this is taking far more time than would have been taken by NIST simply clarifying one way or the other.

---D. J. Bernstein

--