From: D. J. Bernstein <djb@cr.yp.to>
Sent: Monday, September 21, 2020 5:50 AM
To: pqc-comments
Cc: pqc-forum
Subject: ROUND 2 OFFICIAL COMMENT: NTRU Prime
Attachments: signature.asc

I'm writing on behalf of the NTRU Prime team, as a public response to NIST's request for a summary of expected changes in round 3.

NTRU Prime is a small lattice system. Subject to this constraint, our primary objective is to eliminate unnecessary complications in security review. We correctly predicted that such complications would lead to security failures in NISTPQC lattice submissions. We evaluated a variety of trapdoor functions from this perspective before submission, again during round 1, and again during round 2.

On this basis we have once again decided against decryption failures; modules; errors; and all other changes that we have considered to our family of trapdoor functions. We therefore plan to submit the same family of trapdoor functions in round 3. NTRU Prime will therefore have an unchanged family of trapdoor functions throughout round 1, round 2, and round 3.

Our CCA conversion includes various hashing safeguards, some already in round 1 and some added in round 2. These safeguards cost 32 bytes in ciphertext size and a considerable fraction of our CPU time. However, even with these safeguards, NTRU Prime often outperforms other small lattice KEMs, as the following references show:

https://cr.yp.to/papers.html#paretoviz
https://cr.yp.to/talks/2019.08.23-1/slides-djb-20190823-1-paretoviz-4x3.pdf
https://bench.cr.yp.to/results-kem.html#amd64-hiphop
https://github.com/mupq/pqm4/blob/master/benchmarks.md

More importantly, the costs of our hashing safeguards are negligible in applications. We plan to submit the same CCA conversion in round 3.
NTRU Prime will thus be fully compatible between round 2 and round 3, when users choose the same parameters.

Regarding parameter selection, we are concerned that pre-quantum Core-SVP levels 2^100, 2^106, and 2^111, proposed for category 1 for Dilithium, NTRU, and Kyber respectively, will turn out to be inadequate against generic lattice attacks. We will not add dimensions below our
653 (pre-quantum Core-SVP 2^129). We recommend our original dimension
761 (pre-quantum Core-SVP 2^153) for an extra security margin.

We have seen various requests for larger dimensions, even larger than our dimension 857 (pre-quantum Core-SVP 2^175). To accommodate these requests and prevent any accusations of a lack of flexibility, we plan to add some larger dimensions as a supplement to our current dimensions.

We have also considered adding intermediate parameter sets to further illustrate our flexibility, showing that NTRU Prime offers even larger advantages in Core-SVP under various size limits compared to, e.g., Kyber. The call for proposals explicitly allowed multiple parameter sets per category. However, NIST has now made an announcement that seems to discourage "too many parameter sets".

The rest of this message is regarding the problems caused by NIST's unstable definitions of security categories. These are problems for the NISTPQC process broadly, not just for NTRU Prime.

The call for proposals specified AES-128 key search as a "floor" for category 1 in "all metrics that NIST deems to be potentially relevant to practical security". The call for proposals similarly specified floors for other categories. However, this is not a clear and stable category definition unless the metrics are clear and stable.

As an illustration of how much impact metrics have, there is a 40-year literature studying metrics for realistic large-scale two-dimensional models of computation. Standard theorems---see, e.g.,

https://www.eecs.harvard.edu/~htk/publication/1981-jacm-brent-kung.pdf

---imply that these metrics assign 50% higher asymptotic exponents to large-integer multiplication, large-array sorting, etc. than a "gates"
metric does. (Analogous three-dimensional metrics studied in, e.g.,

https://link.springer.com/article/10.1007/BF01744565

are for machines that appear far more difficult to build than quantum computers, and still have 33% higher exponents than a "gates" metric.) Any attack that has large-scale sorting as a bottleneck is affected by this, whereas AES-128 key search is not.

The call for proposals highlighted "classical gates" and "quantum gates"
(with limited depth) as metrics. However, NIST is not requiring lattice submissions to meet the "classical gates" floor. (See examples below.)

NIST also has not defined a replacement metric for submissions to use.
All lattice submissions have Core-SVP evaluations, but AES-128 does not.
Core-SVP is not a metric for the cost of computation: it is a mechanism for claiming security levels (in an undefined metric) specifically for lattices. Ray Perlner's message dated 9 Jun 2020 15:39:09 +0000 stated "we feel that the CoreSVP metric does indicate which lattice schemes are being more and less aggressive in setting their parameters", but the mapping from Core-SVP evaluations to categories remains undefined.

NIST IR 8309's handling of categories is not consistent across lattice submissions. Consider the following examples from three different
submissions:

(P80) Category 3 for pre-quantum Core-SVP 2^153; 153 is 80% of 192.

(P78) Category 1 for pre-quantum Core-SVP 2^100; 100 is 78% of 128.

(P71) Category 3 for pre-quantum Core-SVP 2^136; 136 is 71% of 192.

P78 is the lowest of these three examples in Core-SVP, and P71 is the lowest in Core-SVP as a percentage of the AES key size for the category.
However, NIST's wording was strikingly more negative for P80 than for
P78 and P71:

(P80) "quite aggressive compared to most of the other submissions
targeting the same security categories"; need to study "whether
they actually meet their claimed security categories";

(P78) "lowest CoreSVP security strength parameter set of any of the lattice schemes still in the process"; need more study on "understanding the concrete security";

(P71) "lower CoreSVP complexity than many of the other schemes targeting the same security strength categories"; need to "understand exactly ... bit security strengths".

Notice, e.g., that NIST asks whether P80 "actually" meets its "claimed" security category, while NIST does not ask the same question regarding P78 or P71.

If NIST were applying the "classical gates" metric then none of P80, P78, and P71 would be able to confidently claim these categories. For example, the uncertainties in Core-SVP seem very unlikely to turn Core-SVP 2^100 into 2^143 "classical gates". Most of the remaining lattice submissions (at least Dilithium, NTRU, Kyber, and NTRU Prime; perhaps also SABER after the announcement that SABER's security levels were miscalculated) would have to adjust their category assignments.
Even worse, some of these submissions (at least Dilithium, NTRU, and Kyber) would have to remove some previously proposed parameters, which seems contrary to the idea of being ready for standardization.

All of these submissions argue, with varying levels of detail and references, that the "classical gates" metric underestimates the actual cost of known attacks. NIST seems receptive to the idea of using a more realistic metric, but has taken four years to post its "preliminary thoughts" on the realism of several different metrics. It is not clear what metrics NIST will end up defining, and it is not clear how long NIST will take to settle on the definitions. What is clear is that NIST has not applied the categories consistently, as illustrated by NIST IR 8309 assigning P80 more negative wording than P78 and P71.

The different wording regarding P80, P78, and P71 appears to have translated into different action, and this seems particularly important for NIST's handling of NTRU Prime. As context, NIST IR 8309 describes finalists in general as

> the most promising to fit the majority of use cases and most likely to be ready for standardization soon after the end of the third round.

We have shown that NTRU Prime fits practically all use cases. As far as we can tell, beyond general concerns about the safety of lattice-based cryptography and about the safety of all small lattice proposals, NTRU Prime is ready for standardization now with our existing parameter sets.
The only negative comments that NIST IR 8309 made regarding NTRU Prime were regarding parameter sets. Specifically, NIST seemed to criticize

* NTRU Prime's assignment of pre-quantum Core-SVP 2^153 to Category 3 (this is exactly P80 above),

* NTRU Prime's assignment of _post-quantum_ Core-SVP 2^159 (pre-quantum Core-SVP 2^175) to Category 4 (this has a larger security margin than P80),

* NTRU Prime's assignment of _post-quantum_ Core-SVP 2^117 (pre-quantum Core-SVP 2^129) to Category 2 (this has a larger security margin than P80), and

* NTRU Prime's "narrower range of CoreSVP values" (our understanding now is that this wasn't a negative comment but merely a request for larger parameters going forward).

Meanwhile various lattice submissions with objectively more dangerous parameter selections were given less critical wording by NIST and were selected as finalists. We see no explanation for why NIST treated P78 and P71 in those submissions more gently than P80 in NTRU Prime.

An application limited to 1024 bytes for keys and plaintexts reaches Core-SVP 2^129 with NTRU Prime's proposed parameters and nothing better than 2^111 with Kyber's proposed parameters. 2^129 is higher security relative to category 2 than 2^111 relative to category 1, and obviously higher security on an absolute scale. NIST's report did not acknowledge this security advantage of NTRU Prime.

We are concerned that the lack of clear, stable, consistently applied category definitions will be used in the continuation of NISTPQC to again make NTRU Prime's parameter choices artificially sound worse than more dangerous parameter choices in other submissions. If we try to reduce this risk by downgrading (e.g.) 2^129 to category 1, while Kyber is allowed to remain in category 1 with just 2^111, then NTRU Prime will be unfairly punished in performance comparisons.

We request that NIST issue clear and stable definitions of the metrics used to define NIST's security categories. At this point in the NISTPQC process, clarity and stability are more important than the exact level of realism. Beyond the floor for the categories, one can reasonably argue that users should take higher Core-SVP levels for all lattice submissions in light of continued advances in lattice attacks; but NIST should handle this in a way that is fair to all submissions. As soon as the evaluation criteria are made clear, we will be happy to adjust our category assignments accordingly.

---Dan

From:	D. J. Bernstein <djb@cr.yp.to>
Sent:	Sunday, September 27, 2020 4:43 AM
To:	pqc-forum
Cc:	pqc-comments
Subject:	ROUND 2 OFFICIAL COMMENT: NTRU Prime
Attachments:	signature.asc

This message has three questions for NIST.

In https://www.youtube.com/watch?v=CBGX1OMzN1o&t=37m55s a few days ago, NIST stated "We're we're still uh have some some questions about NTRU Prime" but didn't elaborate. What are NIST's questions about NTRU Prime?

The late notification and lack of information are problematic. NIST has asked for round-3 tweaks by 1 October, which is just a few days from now. Did I miss some NIST publication listing NIST's questions?

I see only one part of NIST IR 8309 that can be understood as a question about NTRU Prime: namely, "whether they actually meet their claimed security categories will need to be determined" regarding the parameter choices. Meanwhile NIST did not ask the same question regarding other submissions that have objectively more dangerous parameter selections.
The NTRU Prime team email dated 21 Sep 2020 11:49:53 +0200 gave examples of this phenomenon. Procedurally, seeing the issue raised by NIST was a prerequisite for responding to it (and the difficulty of responding was exacerbated by the lack of clarity regarding NIST's "categories").

This section of the talk appeared to be presenting NIST's rationale for selecting lattice finalists and lattice alternates. It would thus seem that the existence of NIST's "questions" regarding NTRU Prime played a role in this. Why didn't NIST IR 8309 say this and provide the list of questions?

---Dan (speaking for myself)

From: Moody, Dustin (Fed)
Sent: Monday, September 28, 2020 5:15 PM
To: D. J. Bernstein; pqc-forum
Cc: pqc-comments
Subject: Re: ROUND 2 OFFICIAL COMMENT: NTRU Prime

Dear Dan,

Thanks for your last two official comments on NTRUprime. We'll try to address your questions.

As noted, in my talk at PQCrypto I mentioned that NIST had some questions regarding NTRUprime. In the chat which followed the session, we were asked the same question you asked, i.e. what questions do we have about NTRUprime? We answered that they mainly deal with algebraic cryptanalysis of lattice KEMS which exploit the structure of the ring that they choose, as well as what we wrote in our write-up of NTRUprime in NISTIR 8309. In the chat we also had a few more comments. You were active on the chat, so we assumed you saw our answers. I wasn't trying to bring up anything last-minute. Indeed, some of this has been discussed via email between NIST and NTRUprime back in August. We will expand a bit more below.

While your comment (from Sept 21) focuses on the parametrization of NTRUprime, this was not a major reason for assigning NTRUprime as an alternate rather than a finalist. In particular, we felt that given known facts regarding lattice cryptanalysis, the cyclotomic structures used by the lattice finalists are more researched, and should by default, be considered better understood. In our view, the submitters of NTRUprime have voiced concerns about cyclotomic lattices, but have not proposed any concrete attack against them or any strong argument that NTRUprime's ring choice would rule out a similar attack. We are open to the possibility that there is such an attack, as we are open to the possibility that NTRUprime's ring choice is susceptible to an attack from which cyclotomics are immune. But, without a strong argument that such an attack is ruled out by NTRUprime's ring choice (this seems to us difficult to provide without seeing the hypothetical attack first,) any attack on cyclotomics would undermine our confidence in structured lattices overall. As such, NTRUprime's most probable path to standardization involves two major research results.

1) An attack undermining NIST's confidence in the choice of algebraic structures used by Kyber, NTRU, and Saber

2) The establishment of a strong theoretical barrier for attacks to be extended to the NTRUprime ring.

As the timeline for this path to standardization is by necessity longer than likely paths to standardization for the other finalists, after much deliberation and debate we classified NTRUprime as an alternate. On rereading the report, I realize we did not make this reasoning as spelled-out as it could have been, and for that we apologize.

It should also be noted that the sheer number of structured lattice KEMs in the process always meant we would have to make hard cuts in choosing finalists and alternates. In contrast, as an alternate, NTRUprime is still very much in the process. The same can not be said for other very strong submissions in the structured lattice KEM category that were eliminated from consideration at the end of the 1st and 2nd round of our process.

The fact that there is a lot of discussion in the report about parameter choices and security levels is not because it has factored heavily into our selections thus far, but rather with the intent that, going forward, all submissions in the 3rd round will have the opportunity, through tweaks, to minimize the chances that we choose not to standardize a scheme that we would have, had they chosen other parameters. The issue that the range of parameters offered by the various lattice submissions is highly variable when assigned to NIST security categories was in fact raised on the pqc-forum late in our selection process by you in late June. This was after we had already settled on our list of finalists and alternates, but before our 2nd round report was published. We agreed that it could cause problems going forward, so we highlighted it in our report. The wordings we used varied for substantive reasons (e.g. NTRU's choice to highlight two different computational cost models, and the fact that failing to meet category 1 would result in us discarding a parameter set, while failing to meet a higher level could simply mean changing how it is labeled in our standard.) But the wording also varied due to the personal idiosyncrasies of the writing styles the 13 different authors of the NIST report. If you think subtle differences in wording are any indication of how the NIST team as a whole will evaluate the candidates at the end of the 3rd round, 12 to 18 months from now, independent of any tweaks made by the submitters, and any subsequent analysis that may come to light, you are reading too much into the tone here.

Finally, you request further clarification than we have already given regarding how we intend to assign security levels. We feel we have given about as much as we can give without prejudging scientific questions to which we do not know the answer. We have already stated that gate count in the RAM model exceeding that of brute force AES key search or brute force SHA collision search as appropriate will be taken as strong a priori evidence of meeting any of the NIST categories. If you think more aggressive parameters than are justified via the RAM model are appropriate for meeting a given security target, you will need to make a strong argument that convinces us. We've given reasons why previous arguments have not completely convinced us, so you should take that into account. We think there is a real possibility such arguments could convince us -- we just haven't been completely convinced by any yet. Likewise, while variations in wording about how we raise questions about a particular parameter set are likely to be uninformative, we think we were fairly consistent in which lattice parameter sets we did and did not raise concerns about when claiming each security strength category. Parameter sets we didn't raise concerns about are probably ok for their claimed security levels, barring developments in cryptanalysis. If you think we've missed something in this regard, please let us know, because it likely indicates a relevant research result we were not aware of at the time of writing our report.

In any event, we think the public statements we've made on the forum and in our report are sufficient for any submission team working in good faith to determine what parameter sets will be uncontroversial, controversial and unacceptable for the claimed security levels given the current state of knowledge. Keep in mind that extra care is warranted for the lowest (and perhaps highest) security levels offered by a submission. For example, while a controversial assignment of category 1 runs the risk of the parameter set in question not being standardized, a controversial assignment of category 3 probably just runs the risk of the parameter set in question being downgraded to category 2, or at worst, category 1. Stakeholders who care about category 3 can then take our assignment into account when deciding which NIST standardized parameter set to use. A controversial assessment of category 5 runs the risk that the submission will not meet the needs of any users who actually want category 5. We reiterate what we've been saying since the beginning of the PQC process that, barring future cryptanalysis, category 1 parameters are probably enough to thwart any purely computational attack in the near to medium term future, and category 3 is almost certainly enough.

We do try to respond promptly to questions brought to us, but it does take a little bit of time for our team to discuss and draft a response. The updated specifications and implementations are due on October 1st. We've tried to be flexible throughout the process. Teams can always contact us to ask for a bit more time if they feel they need it.

Dustin and Ray
The NIST PQC team

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Friday, September 3, 2021 4:53 PM
To: pqc-comments
Cc: pqc-forum
Subject: ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments: signature.asc

Two years ago an NTRU Prime update talk announced the new "factored"
NTRU Prime software, a wrapper around "modules with separate tests and optimizations". I gave a talk today announcing, among other things, computer verification for most of these modules that the existing "avx" implementation produces the same output as the existing "ref"
implementation for _all_ possible inputs:

https://cr.yp.to/talks.html#2021.09.03

This is a big step towards full verification of the optimized NTRU Prime software. Next steps are matching "avx" to "ref" for the other modules (notably multiplication, where another tool has gotten through some important parts of the code but not yet everything) and matching the C code to the Sage reference code.

The "saferewrite" tool used for this verification has a broad range of applicability beyond NTRU Prime. The first example in the talk is how saferewrite catches both of the array-comparison problems that were announced in the official Frodo software. However, to enable this analysis, I had to define a Frodo array-comparison module and write reference code for that module, so that saferewrite could compare the official code to my reference code. This wasn't a big deal since this particular module is so simple, but an analogous analysis for larger components of Frodo (short of taking the entire KEM as a monolith!) would require additional work to write reference code for those modules.
For NTRU Prime, this work is already done.

For each module, saferewrite compiles each implementation with clang -O1 and with gcc -O3, uses the angr symbolic-execution toolkit to convert each binary into unrolled code in a much simpler language, and uses the
Z3 theorem prover (via angr's claripy) to verify equivalence or find a counterexample. The automatic equivalence chains look like this (although this pattern isn't optimal in general):

```
   opt clang -O1 = ref clang -O1 = avx clang -O1
                ||
     opt gcc -O3 = ref gcc -O3 = avx gcc -O3
```

There could be compiler bugs affecting outputs, but to evade detection these bugs would have to have the same effect on every node in the diagram simultaneously. (It would also be possible to hook a direct Python-to-the-simpler-language conversion into the picture.) There could be unrolling bugs, but saferewrite also runs the binaries on some random inputs (plus all-0 and all-1 to make sure every bit is touched) and checks that these match the unrolled code; also, angr has been heavily exercised in a variety of reverse-engineering applications. There could be bugs in saferewrite itself, but reviewing saferewrite is a much smaller task than reviewing a ton of optimized post-quantum code.

The examples supplied in the saferewrite package include deliberately buggy code to exercise saferewrite's tests, in particular producing 16 analyses printing "differentfrom" counterexamples (which I've checked by hand), providing some evidence that saferewrite is working as desired.
Some of these bugs are also found by random tests, but some aren't. More advanced fuzzing can do better than random tests but has no hope of finding typical cryptographic overflow bugs.

Seeing C code working with two compilers doesn't mean that the same code will work with further compilers, but if analyses are fast enough then it's realistic to re-apply the analyses whenever the compiler changes. I tried the saferewrite analysis of 107 implementations of 27 functions, times 2 compilers, on a dual EPYC 7742; it finished in 8 minutes of wall-clock time, using 20 cores on average, using under 200GB of RAM.

I'm filing this as an OFFICIAL COMMENT regarding NTRU Prime because it's directly relevant to the official NISTPQC evaluation criteria, notably the following:

> The algorithms can be implemented securely and efficiently on a wide variety of platforms, including constrained environments, such as smart cards.

Various NISTPQC submissions have provided fast AVX2 software, fast M4 software, etc., but the primary evidence for "securely" is that the software is constant-time. (I'll skip discussion of the broken masked implementations.) The problem is that the same optimizations add massive complexity to the software, and this complexity is a security threat:

* https://arxiv.org/abs/2107.04940 studied the vulnerabilities announced between 2010 and 2020 in eight well-known cryptographic libraries, and found 73 vulnerabilities in the cryptographic computations, including 11 known to be exploitable ("severe"), along with "evidence of a strong correlation between the complexity of these libraries and their (in)security". (There were also hundreds of further bugs, such as buffer overflows.)

* Post-quantum software is newer, more complicated, and much harder to thoroughly review. Superficial reviews of post-quantum software have caught one devastating bug after another; the only reasonable prediction is that more serious reviews will find many more bugs.

Is it reasonable to say that an algorithm "can be implemented securely and efficiently" if fast implementations are so complex that the experts are getting them wrong? If the answer is pointing to an implementation and saying "No bugs are known in this implementation", then why should we think that the code is correct, rather than thinking that security reviewers are overloaded and that this answer is pure selection bias?

There's stronger evidence for "securely and efficiently" when optimized modules are verified to match much simpler reference implementations.
Covering more modules will further strengthen this evidence.

Another relevant NISTPQC evaluation criterion is the following:

> Factors that might hinder or promote widespread adoption of an algorithm or implementation will be considered in the evaluation process, including, but not limited to, ...

The availability of modularized implementations, and the availability of verification tools applicable to some of those modules, certainly help promote widespread adoption of those implementations and the algorithm.

---Dan (speaking for myself)

From: pqc-forum@list.nist.gov on behalf of Wrenna Robson <wren.robson@gmail.com>
Sent: Saturday, September 4, 2021 5:51 AM
To: pqc-forum
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

Thank you for this, Dan - really interesting. I am hopeful of having something to share in the next couple of months on my own verification and validation efforts with Classic McEliece using SAW/Cryptol, and certainly much of your thinking here seems to align with conclusions and thoughts I've reached.

(Increasingly I think a hybrid approach, using multiple toolchains that complement each other's weaknesses, with a clear analysis and synthesis of the truth-claims that each makes. is the way forward - though this mean, of course, you have more tools that you need to validate and verify...)

- Wrenna

On Fri, 3 Sept 2021 at 21:53, D. J. Bernstein <djb@cr.yp.to> wrote:

> Two years ago an NTRU Prime update talk announced the new "factored" NTRU Prime software, a wrapper around "modules with separate tests and optimizations". I gave a talk today announcing, among other things, computer verification for most of these modules that the existing "avx" implementation produces the same output as the existing "ref" implementation for _all_ possible inputs:
>
> https://cr.yp.to/talks.html#2021.09.03
>
> This is a big step towards full verification of the optimized NTRU Prime software. Next steps are matching "avx" to "ref" for the other modules (notably multiplication, where another tool has gotten through some important parts of the code but not yet everything) and matching the C code to the Sage reference code.
>
> The "saferewrite" tool used for this verification has a broad range of applicability beyond NTRU Prime. The first example in the talk is how saferewrite catches both of the array-comparison problems that were announced in the official Frodo software. However, to enable this analysis, I had to define a Frodo array-comparison module and write reference code for that module, so that saferewrite could compare the official code to my reference code. This wasn't a big deal since this particular module is so simple, but an analogous analysis for larger components of Frodo (short of taking the entire KEM as a monolith!) would require additional work to write reference code for those modules. For NTRU Prime, this work is already done.
>
> For each module, saferewrite compiles each implementation with clang -O1 and with gcc -O3, uses the angr symbolic-execution toolkit to convert each binary into unrolled code in a much simpler language, and uses the Z3 theorem prover (via angr's claripy) to verify equivalence or find a counterexample. The automatic equivalence chains look like this

From: pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent: Wednesday, September 8, 2021 10:16 PM
To: pqc-forum
Cc: pqc-comments
Subject: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments: complaint-re-apon.pdf; signature.asc

Each round-3 submission team was given a 15-minute slot at the NIST conference three months ago to present updates for, and field questions from, an online audience of about 300 people, of course including NIST.

During the NTRU Prime talk, Dr. Apon posted the text quoted below to the Slack channel for the conference, publicly accusing me of professional misconduct. Specifically, he accused me of initiating private contact with NIST so as to provide false information to NIST regarding the timing of an upcoming announcement relevant to NIST's ongoing decisions.
However:

* The contact was requested by NIST.

* The false information that Dr. Apon attributes to me is a fabrication by Dr. Apon.

Here's Dr. Apon's text:

> 9:38 PM
> Daniel Apon @djb a quick, gentle question:
>
> Regarding cyclotomic-based attacks: 13-14 months ago (just before the end of the 2nd Round), you privately emailed NIST PQC to suggest that you had a new attack paper (in the line) against cyclotomics that was coming out in 2-3 months. Of course, we are very eager to hear any progress along this line, even attacks that provide progress on this line, even if they don't break a cryptosystem outright (but might threaten cryptosystems in the future). However, no paper came out. We invited you to give a public 3rd Round Seminar talk on this issue in the Fall, and you ended up giving a talk at the Seminar Series in January 15 of this year. The talk presented a variety of algebraic and mathematical background that was quite interesting, but didn't suggest a clear attack vector. At the time, I suggested that you finish the paper and submit the attack paper to this conference. We didn't receive such a submission. Given this background of no attack progress against cyclotomics since the beginning of the pandemic (after claiming at least an epsilon of progress would be coming in "a month or months" well over a year ago), how would you characterize your progress in making a single epsilon of progress in attacking cyclotomic structures in lattice-based cryptography?

I replied quickly on the Slack channel to the beginning of this---

> please don't misrepresent the history. nist specifically asked to be
> informed regarding ongoing projects, and i answered.

---but then, reading further in Dr. Apon's text, I found one outright fabrication after another, and responded accordingly:

> i'll post an apon fact check to pqc-forum in due time. in the
> meantime, happy to answer honest questions.

I promptly contacted Dr. Apon:

> I'm writing to request a retraction of the "question" that you issued
> during my talk. I expect the retraction to include a clear and
> specific acknowledgment that you fabricated the "coming out in 2-3
> months" part, and that this part plays a critical role in your
> "question". I'll give you 1 week before escalating.

I sent another message to Dr. Apon shortly after that:

> Appendix: I also expect the retraction to include the same clear and
> specific acknowledgment regarding your fabrication of the "month or
> months" part, which plays the same role in your "question".

A week later I filed a complaint with NIST, reviewing the relevant facts in detail and comparing them to Dr. Apon's accusation, and also raising the obvious procedural points.

NIST replied a week after that, admitting that the timing of Dr. Apon's "question" was "not proper", but did not address any of the gaps that I had identified between the facts and what Dr. Apon wrote. I asked two clarification questions, which were never answered.

A few weeks later I further escalated the complaint within NIST. NIST replied after a week, again not addressing the factual dispute. Two weeks later I asked what options were available within NIST for resolving the factual dispute. There was no reply.

Naturally, I cc'ed Dr. Apon on the complaint and every subsequent message. He has had three months to tell me how exactly he believes I'm getting the facts wrong or missing something that justifies his accusation. He has not taken this opportunity.

I am now posting the fact check as promised. See attached for a copy of the complaint that I filed with NIST. This incident appears to be part of a larger problem with continuing impact on NIST's evaluation of NTRU Prime, so I'm filing this as an OFFICIAL COMMENT regarding NTRU Prime.

---Dan (speaking for myself)

Formal complaint regarding 8 June 2021 incident

2021.06.15
Daniel J. Bernstein

Executive summary. A week ago Dr. Daniel Apon from NIST publicly accused me of professional misconduct. Specifically, he accused me of initiating private contact with NIST so as to provide false information to NIST regarding the timing of an upcoming announcement relevant to NIST's ongoing decisions. However:

- The contact was requested by NIST.
- The false information that Dr. Apon attributes to me is a fabrication by Dr. Apon.

All of this contact was official email exchanged with `dustin.moody@nist.gov`. Anyone at NIST with access to the email can see that it disproves Dr. Apon's accusation.

Complaint overview. There are three independent prongs in this complaint:

- Content: Dr. Apon issued a false accusation.
- Procedures: Dr. Apon did not follow proper procedures regarding the choice of venue and timing for issuing this accusation.
- Intent: Dr. Apon's choice of content, venue, and timing for this accusation were malicious.

Dr. Apon's false accusation was published for an audience of more than 300 attendees of a NIST conference. It's not clear at this point how much work will be required to address the damage. It is clear, however, that at a minimum Dr. Apon must issue a prompt retraction of his accusation, including clear acknowledgments of

- the specific fabrications described below and
- the critical role that two of these fabrications play in Dr. Apon's accusation.

For obvious reasons, I request an opportunity to review and approve the text and venue for the retraction before the retraction is published.

Dr. Apon has already had an opportunity to see and respond to the core of each prong of this complaint:

- Regarding content, I sent email to Dr. Apon dated 8 Jun 2021 21:53:21 +0200 quoting a specific part of his accusation, identifying this as a fabrication playing a critical role in his accusation, and requesting a retraction of the accusation. This was approximately 15 minutes after his accusation appeared. I sent followup email dated 8 Jun 2021 23:54:44 +0200 identifying a further fabrication.
- Regarding procedures, I sent email dated 9 Jun 2021 00:03:20 +0200 to Dr. Dustin Moody, cc'ing Dr. Apon, to ask whether Dr. Moody knew in advance that Dr. Apon was going to do this. (Dr. Moody said he didn't know in advance.) This was only a procedural question, not a complaint, but it included a summary of why Dr. Apon's choice of venue and timing were problematic.
- Regarding intent, I followed up in the same venue asking Dr. Apon not to misrepresent the history; saying I would "post an apon fact check to `pqc-forum` in due time"; and saying I was "happy to answer honest questions". Dr. Apon saw this and replied with an "eyes" emoji.

The only response I've seen from Dr. Apon is (1) generically claiming sincerity and (2) bringing his supervisors into the discussion. Dr. Apon has not addressed the specific fabrications that I identified, or the procedural issues. I conclude that it is appropriate for me to escalate to Dr. Lidong Chen at this point.

The NIST Post-Quantum Cryptography Standardization Project (NISTPQC). In 2016, NIST called for public submissions to its Post-Quantum Cryptography Standardization Project. NIST created a public mailing list, `pqc-forum`, for discussions of submissions.

I had coined the term "post-quantum cryptography" in 2003, and I have various papers on the topic. I joined a few teams preparing submissions.

NIST set a submission deadline of 30 November 2017; posted 69 "complete and proper" submissions on 21 December 2017; held a First PQC Standardization Conference on 11–13 April 2018; announced its selection of 26 submissions on 30 January 2019 for further consideration; held a Second PQC Standardization Conference on 22–24 August 2019; announced its selection of 15 submissions on 22 July 2020 for further consideration; and held a Third PQC Standardization Conference online last week, 7–9 June 2021.

After the submission deadline, Dr. Apon asked me for a faculty job (email dated 18 Dec 2017 13:55:44 -0800). I immediately filed his application as "reject without notification". He appears to have taken this personally, and appears to be abusing his position at NIST. Given his failed job application, he should have recused himself from every action and decision related to my NISTPQC submissions, but he did not do so.

NIST's pattern of encouraging private input in NISTPQC. NIST ran what it called a "survey" at the Second PQC Standardization Conference. This "survey" was not limited to any specific questions: it was an open-ended request for public inputs regarding all aspects of NISTPQC. NIST posted anonymized versions of some of the inputs: e.g., one comment advocated prioritizing "security" and "maybe size" while ignoring "performance". NIST promised to take all of the inputs into account, not just the ones it had posted.

There were subsequent examples of NIST requesting private input. For example, in email to `pqc-forum` dated 30 Oct 2019 15:38:10 +0000 (2019), NIST posted technical comments regarding hybrid encryption modes and asked for feedback "either here on the pqc-forum **or** by contacting us at `pqc-comments@nist.gov`" (emphasis added).

In July 2020, during the second round, I finally realized what was wrong with this picture. I tweeted the following on 22 Jul 2020 13:01 GMT (`https://twitter.com/hashbreaker/status/1285922808392908800`):

> After NIST's Dual EC standard was revealed in 2013 to be an actual (rather than just potential) NSA back door, NIST promised more transparency. Why does NIST keep soliciting private #NISTPQC input? (The submissions I'm involved in seem well positioned; that's not the point.)

This happened to be about 2 minutes before NSA sent its first message to `pqc-forum`. NIST announced round 3 some hours later, and eventually admitted coordination with NSA, although the exact extent of the coordination remains unclear to the public.

Applicable transparency principles. NSA has a long history of manipulating standards. Before NIST (at the time NBS) issued its first cryptographic standard, NSA established an internal policy goal of making this standard "weak enough to still permit an attack". See `https://cr.yp.to/papers.html#competitions` for the full quote and references.

News reports in 2013 indicated that NSA had successfully manipulated NIST into issuing a Dual EC standard backdoored by NSA. In response, the NIST VCAT Dual EC report (see `https://www.nist.gov/system/files/documents/2017/05/09/VCAT-Report-on-NIST-Cryptographic-Standards-and-Guidelines-Process.`

`pdf`) strongly encouraged NIST to run "open competitions" and to follow various transparency principles. As Bart Preneel put it in that report, transparency includes "version control on all documents from an early stage, a full documentation of all decisions, and clear processes for the disposition of each and every comment received". As Steven B. Lipner put it in the same report, transparency includes being open about "what steps were followed, what authorities were consulted or reviews sought, what comments were received, and what actions or resolutions reached".

For NISTPQC, the call for submissions stated that NIST would "perform a thorough analysis of the submitted algorithms in a manner that is open and transparent to the public", and that NIST's decisions would also take into account "public comments" received in response to the submissions. There was nothing in the call indicating that NIST would also take into account *private* comments, from NSA or from anyone else.

In fact, NIST's analysis within NISTPQC is not transparent to the public. For example, despite the NIST VCAT Dual EC report, NIST is not open about which comments it is receiving from which sources, and does not have transparent processes for the disposition of comments.[1] Beyond the lack of basic transparency, NIST refuses to designate NISTPQC as a "competition", despite the VCAT report having strongly encouraged open competitions; it appears that NIST decided that it didn't want to follow the additional rules that would have been triggered by the word "competition".

On 28 August 2020, NIST gave a talk explaining "NIST's decision process for Round 3", with information that had not appeared in NIST's report a month earlier. There was no public announcement of the talk. I filed a FOIA request (email dated 1 Sep 2020 17:10:24 +0200), in response to which NIST publicly (email dated 1 Sep 2020 16:11:44 -0700) described the talk as having been "given to the University of Maryland Crypto Reading Group" and posted the slides.

The FOIA request was actually for "the list of people that NIST emailed to give access to the talk". The response eventually showed that NIST had invited *some* round-3 submitters—not from the University of Maryland—to attend the talk and the subsequent Q&A session. This included, for example, members of submission teams for six lattice submissions: Dilithium (2×), Falcon (2×), Frodo, Kyber (3×), NTRU (2×), and SABER. It did not include any members of the submission team for the only other lattice submission, NTRU Prime.

Examples of consequences of the lack of transparency in NISTPQC. One aspect of NIST's round-3 report that surprised me was a new provable-security argument that "the security of NewHope is never better than that of KYBER". This was almost half of NIST's text in that document regarding NewHope, and appeared to play a prominent role in NIST's decision to eliminate NewHope.

The reason this surprised me is that it was contradicted by the literature on the topic. What NIST wrote has *some* overlap with a proof in the literature, but there are also clear and irreconcilable gaps. NIST downplayed these gaps as "a few minor caveats", but the simple fact is that the proof never said what NIST claimed.

I filed an official comment (email dated 23 Jul 2020 11:32:30 +0200) saying "I don't believe the argument. I'm filing this comment to request that NIST spell out the argument in more detail for public review." Various cryptographers followed up, pointing to some

[1] Long before July 2020, I had heard from colleagues about submitters having detailed private discussions with NIST. I had dismissed these as isolated rumors. Later, when I asked what talks had been given to NIST before round 3, Dr. Apon wrote (email dated 11 Nov 2020 17:29:56 +0000) that "we didn't have any talks that I recall in the prior rounds".

of the reasons that “there is no known reduction between Kyber and NewHope”. I followed up (email dated 25 Jul 2020 10:36:57 +0200) pointing to even more obstacles to making NIST’s argument work, and then pinpointed how a series of technical errors would lead to exactly what NIST had written. I concluded with a procedural objection:

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

NIST replied to my objection by reviewing what the theorem says, admitting one specific “caveat”, and then switching to *different* arguments against NewHope, arguments that had not appeared in the report.

Another aspect of NIST’s round-3 report that surprised me was the following. The report said that NIST “strongly encourages” submissions to provide a parameter set “that meets category 5”, and then criticized some submissions that provided at most category 4. The reason this surprised me is that the call for submissions had said something different: NIST “recommends” that submissions provide a parameter set “above category 3”. The report didn’t mention that NIST had changed from “recommends … above category 3” to “strongly encourages … category 5”, never mind explaining why the change had happened. When I publicly objected, one of NIST’s attempted defenses was the following remarkable admission (email dated 31 Jul 2020 14:42:02 +0000):

> Throughout the process we’ve been in dialogue with various teams as they have adjusted parameter sets.

I responded as follows (email dated 2 Aug 2020 11:50:26 +0200):

> You’re talking about *private* discussions that NIST has had with various teams? This is supposed to somehow be a replacement for having a change in evaluation criteria proposed and discussed in public? Wow. When did NIST announce that submitters were expected to use this private source of information rather than relying on the public announcements?

There was no reply.

I’ve skimmed the pre-round-3 email that NIST sent to the submission teams that I’m on. There were various administrative messages. There were requests to add certain types of public software documentation; these requests could also have been made public. The only bit that looks to me anything like “dialogue” regarding algorithm desiderata was a clarification question from the SPHINCS+ team:

- NIST’s report at the end of the 1st round said “A possible second-round tweak might involve a more efficient technique for protecting against multi-target attacks.”
- The SPHINCS+ team asked whether NIST was asking for “an LMS variant”.
- NIST wrote “Yes – this is what we were meaning with our comment. It would be great if you could include an LMS variant as one of your tweaks for round 2.”

In retrospect, obviously this clarification should have been public; or, even better, stated more clearly in NIST's public report in the first place. Anyway, none of these teams—including the team for a submission where the largest category proposed at the time was 4—were given any hint that NIST was going to change "recommends ... above category 3" to "strongly encourages ... category 5".

In September 2020 I posted a paper https://cr.yp.to/papers.html#categories observing how easy it was to manipulate NIST's procedures to favor particular submissions (e.g., weaker submissions), and pointing out a variety of transparency failures in NISTPQC. Many of the questions for NIST in that paper remain unanswered today.

NIST's requests for information by 15 April 2020. Let's rewind to the beginning of 2020. By email to `pqc-forum` dated 28 Jan 2020 16:23:39 +0000, NIST set a deadline of 15 April 2020 for input regarding its end-of-round-2 decisions:

> In order to give NIST time to finalize our evaluation and analysis for the candidate algorithms in the 2^nd round, NIST kindly requests that we be notified of new implementations, benchmarks, research papers, cryptanalysis, etc. by April 15^th. After that date, factoring any of that information into our decision-making process may overly tax our resources.

In email to `pqc-forum` dated 26 Mar 2020 19:58:32 +0000, NIST repeated this request. It also asked people to use `pqc-forum` to "announce results, discuss relevant topics, ask questions, etc." I sent email to `pqc-forum` dated 3 Apr 2020 00:15:35 +0200 to recommend a deadline extension:

> My impression is that many people who don't know any COVID-19 victims are nevertheless losing large fractions of their previously expected work time as a result of actions taken to slow the spread of COVID-19. It's not realistic to ask for full work days from, e.g., parents suddenly taking care of kids at home all day.
>
> I'm probably close to the low end of involuntary COVID-19 disruption but this low end certainly isn't zero; also, I've been volunteering some of my own research time to COVID-19 modeling (including scripts online and already a paper a few days ago). I still hope to make the 15 April target date for posting updated NISTPQC benchmarks and the results of some other NISTPQC experiments I've been running, but the big picture makes me think that shifting the timeline would be a good idea.

NIST sent email to `pqc-forum` dated 3 Apr 2020 16:01:41 +0000 as follows (emphasis added):

> If you or anyone else in the community has something important in the works, but don't think it will be done by April 15, **please notify us (by the 15th) with a brief description of the expected results and an estimate of how much longer might be needed**.

Any experienced scientist knows how time-consuming it is to fill out reports describing "expected results" that aren't yet ready to announce. However, given NIST's power over NISTPQC, people involved in the process really don't have the option of ignoring NIST's requests.

In retrospect, it's clear to me (1) that NIST has been continually manipulated by years of private lobbying regarding NISTPQC; (2) that the best response to NIST's requests, already at the 2019 conference, would have been to object to NIST taking private input; and (3) that following NIST's April 2020 request opened me up to ad-hominem attacks as retribution for my procedural objections.[2] But, again, I didn't realize the problem until July 2020.

So, by email dated 15 Apr 2020 02:35:20 +0200, complying with NIST's request, I sent NIST an "overview of what's cooking from my perspective". Other cryptographers sent me copies of various messages that they had sent to NIST. Presumably there were many further messages to NIST that I never saw. There were also dozens of public team announcements on `pqc-forum` around that date.

What the email said regarding timelines. This email dated 15 Apr 2020 02:35:20 +0200 began with an "easy part" listing nine projects (some that I said I was working on, and some that I said I had heard about). This part of the message had various estimates of when announcements would be ready:

- "momentarily" although "maybe not exactly on the 15th" for updated SUPERCOP benchmark results (**2021 update**: this took slightly more than 24 hours);
- "by the end of the year" for verification of "*some* big subroutines" and "*a few* proofs" (**2021 update**: I posted a paper "Verified fast formulas for control bits for permutation networks" in September 2020);
- "doesn't look too hard to finish but has been on the back burner. Please let me know if this is something that you think is high priority" for "a step-by-step proof survey for Classic McEliece" (**2021 update**: NIST never asked me to prioritize this, and it's still on the back burner);
- "about to release" for Cortex-M4 NTRU Prime code (**2021 update**: this code was released a week later);
- "after the updated SUPERCOP Haswell results are online for everybody" for new Haswell NTRU Prime speeds (**2021 update**: this took under 48 hours);
- "this week" for "what I think is fair to describe as a dramatic performance improvement for sntrup761 keygen" (**2021 update**: that week I announced 166000 cycles where previous work was around 800000 cycles).

Notice the wide range between "this week" and "by the end of the year"; saying that this was the easy part of the email doesn't mean that these were easy projects! I also included three pointers to existing announcements that NIST might have missed.

The email continued with a "hard part", namely that "I've been doing a deep dive into lattice security", and listed five "expected results" within this. Kris Gaj had proposed a two-month extension, and it was clear to me that if NIST did this then by reshuffling research time I could get some of the results from the "hard part" online by then. So I wrote that the "hard part" would "clearly benefit from the timeline that Kris Gaj proposed to you". (**2021 update**: NIST didn't extend the deadline, so each part of the research continued on its natural schedule.)

Within the "hard part", the five "expected results" listed in the email were as follows:

- "there's a hybrid attack that reduces the 'Core-SVP security level' of, e.g., Kyber512" (**2021 update**: I spelled out the critical points in email to `pqc-forum` dated 7 Jul 2020 19:06:42 +0200);
- there are "indisputably reasonable 'Ring-LWE' parameters for which switching to 'NTRU' would increase the 'Core-SVP security level' while decreasing the probability of decryption failures" (**2021 update**: the point of this is the looseness of a partic-

[2] For example, in reply to a technical message that I had posted on another topic, Dr. Apon sent email to `pqc-forum` dated 22 Aug 2020 13:27:43 -0700 going far out of his way to mention my "private input" to NIST. NIST has not similarly mentioned the private input it received from other submitters.

ular proof; after checking examples of NIST ignoring previous examples of proof looseness, I decided to put this project on the back burner);

- the "lattices that the NTRU Prime submission presents for attacking 'Ring-LWE' generalize the standard Kannan–Bai–Galbraith lattices to allow more short vectors, and (new result) also outperform the standard lattices" (**2021 update**: I have enough computer experiments to be confident in what I wrote, but analysis and optimization are continuing so as to understand what happens for larger sizes);
- "the huge enumeration speedup announced as work in progress" by another team "also makes a big difference in the enumeration-vs.-sieving cutoffs, especially in realistic models of large-scale computation that account for the costs of memory" (**2021 update**: this analysis ended up in the round-3 NTRU Prime submission in October 2020); and
- there's a "new way to exploit the structure of cyclotomics"—I said this wasn't "big enough to reach the lattices in NISTPQC submissions, but it breaks solidly through barriers claimed in previous work"[3] (**2021 update**: I gave a talk on "Valuations and S-units" in January 2021 explaining the most important number-theoretic background; I have a talk on "S-unit attacks" scheduled for August 2021, and expect that my coauthors will authorize revealing the most important new idea during that talk).

The email did not provide time estimates for any of these five specific items.

I hadn't realized at this point that NIST should have been categorically forbidding private input from the outset. However, given the long history of errors regarding lattice security in particular, I knew that public review was essential in this area. The email closed with a clear warning on this topic:

> I can imagine selection of lattice systems being the hardest decision that NIST is facing, and I can imagine some of these lattice-security issues showing up as part of the decision. **In a non-COVID-19 world, I would have had some of this online already as papers, and then there would be time for the community to comment, which I think is especially important in an error-prone area.**
>
> On the other hand, I also find it easy to imagine that you're planning to avoid tricky issues in this round and make decisions based primarily on undisputed attacks, performance features, simplicity, stability, diversity, etc. This reminds me: has NIST decided how high a weight it's putting on patents in this round?

(Emphasis added.) This was before NIST issued its round-3 report with, e.g., a decision based on a "security of NewHope is never better than that of KYBER" provable-security claim that, as mentioned above, had not been publicly reviewed and did not match what the literature said.

NIST engaged me in followup discussion over the next two weeks. None of my followup email had any timeline predictions.

Content of Dr. Apon's accusation. In a nutshell, Dr. Apon accuses me of having deceived NIST regarding the timing of an announcement regarding cyclotomics. This accusation relies critically on the following two fabrications by Dr. Apon:

[3] Since cyclotomics show up later in this complaint, here's the full quote: "Deepest result, joint work with various people: There's a new way to exploit the structure of cyclotomics, quickly finding a short secret vector from any lattice in a much larger class than handled by previous algorithms. This class isn't big enough to reach the lattices in NISTPQC submissions, but it breaks solidly through barriers claimed in previous work. Top consequence from my perspective: New reason to be scared of cyclotomics."

- Dr. Apon quotes me as telling NIST 13–14 months ago that a cyclotomic announcement would be in “a month or months”. I said no such thing.
- Dr. Apon paraphrases me as saying that a “paper” was “coming out in 2-3 months”. I said no such thing.

Dr. Apon contrasts “coming out in 2-3 months” and “month or months” to the lack of any publication for more than a year. Without Dr. Apon’s fabrications of a much shorter claimed timeline, there would be nothing remarkable about a serious scientific project taking more than a year.

The following paragraphs go step by step through Dr. Apon’s accusation, pinpointing the above timeline fabrications and further fabrications by Dr. Apon. The “9:38 PM” time shown here is in the CEST time zone on 8 June 2021.

> Daniel Apon 9:38 PM @djb a quick, gentle question:

This introduction was followed by more than 1000 characters (216 words) of text (all posted at once as part of the same message), so the reader instantly sees that “quick” is facetious, and presumes that “gentle” is also facetious. This doesn’t make the reader imagine that Dr. Apon’s subsequent claims regarding the history are outright fabrications.[4]

> [Next words in Dr. Apon’s message:] Regarding cyclotomic-based attacks:

The reader interprets further text within this scope.

> [Next words in Dr. Apon’s message:] 13-14 months ago (just before the end of the 2nd Round), you privately emailed NIST PQC

Let’s compare this to the facts:

- NIST set a deadline of 15 April 2020 for input, saying that input after this “may overly tax our resources”.
- Beyond public announcements, NIST specifically asked to be given a “brief description” of any “expected results” and an “estimate of how much longer might be needed”.
- Unlike many other government agencies, NIST refused requests for deadline extensions in light of COVID-19.
- Many people provided input on or around 15 April 2020, the deadline that NIST had set. I was one of these people.
- Three months later, NIST ended the second round with an announcement on 22 July 2020. NIST could have taken longer if it wanted to—there was never any public commitment to this particular date until the announcement happened.

Does Dr. Apon inform the reader that NIST had asked for input by 15 April 2020? No. That many other people also provided input on or around this date? No. That NIST also received private input from many other submitters? No. That NIST received private input from other submitters long before this? No. That my email cautioned NIST regarding the importance of public review especially in the area of lattice security? No.

Yes, 15 April 2020 was 13–14 months ago. But is it correct that this was “just before the end of the 2nd Round”? No. First, “just before” can’t reasonably be understood to include a stretch of three months. Second, like other members of the general public, I didn’t know when round 2 would end—as far as I know, NIST was free to take as much time as it wanted—so saying “just before the end of the 2nd Round” is nonsensical as

[4] If I were to change the title of this document to “A quick, gentle complaint regarding Dr. Apon”, the reader would still take the rest of the document at face value.

a description of what I did in April 2020.

Some readers will know that NIST never publicly scheduled the end of the round until issuing the 22 July 2020 announcement; so, if they think about it for a moment, they'll realize that what Dr. Apon said here is nonsense. However, for a reader who doesn't know this and who takes Dr. Apon's text at face value, the text implies that NIST *had* publicly scheduled the end of the round, and says that I targeted NIST with a surprise just before this. That isn't true.

> [Next words in Dr. Apon's message:] to suggest that you had a new attack paper (in the line) against cyclotomics that was coming out in 2-3 months.

Does Dr. Apon inform the reader that the message actually covered 14 different projects, with many different timelines? No.

The core of what the reader has been told at this point is that I wrote to NIST 13–14 months ago to suggest that a new attack paper on cyclotomics was coming out in 2–3 months. The reader assumes that no such paper has appeared—why else would Dr. Apon start by talking about the timeline and using words like "suggest"?—and concludes that I provided grossly inaccurate information to NIST.

But the timeline information is a fabrication by Dr. Apon. I never gave NIST a timeline for the cyclotomic attacks.

Perhaps Dr. Apon will point to the comment "In a non-COVID-19 world, I would have had some of this online already as papers, and then there would be time for the community to comment, which I think is especially important in an error-prone area" and say that these papers obviously can't have lost more than a few months given that this was a message in April 2020. However:

- "Some" is not "all". Can someone who reads "I would have had some of this online already as papers" claim that this says "I would have had all of this online already as papers"—or select one part X of this and claim that it says "I would have had X online already as a paper"? No.
- Suppose a message is saying, in April 2020, that something was within a few months of done in January 2020, but because of COVID-19 isn't done yet. Can one claim that this message is saying that it will be done in July 2020? No.

The bottom line is that the email never said what Dr. Apon claims it said.

> [Next words in Dr. Apon's message:] Of course, we are very eager to hear any progress along this line, even attacks that provide progress on this line, even if they don't break a cryptosystem outright (but might threaten cryptosystems in the future).

Does Dr. Apon inform the reader that the email had already put this cyclotomic attack into this category, saying that the attack wasn't "big enough to reach the lattices in NISTPQC submissions, but it breaks solidly through barriers claimed in previous work"? No. Dr. Apon suppresses this information, making it sound as if NIST had been given no information regarding the nature of the advance and was merely guessing on its own that the advance wasn't a full break of NISTPQC submissions yet.

Omitting this information is exaggerating what the message said. When the results *are* released, is Dr. Apon going to rely on the same exaggeration to claim that he thought the results were already supposed to be a full break of a NISTPQC submission? Will he express disappointment that the results are "only" breaking some underlying problems? I haven't seen NIST trying to preemptively sabotage announcements that it hears about in advance from other people.

> [Next words in Dr. Apon's message:] However, no paper came out.

It's correct that this project has not released a paper yet.

The "however" is drawing a contrast between (1) this fact and (2) claims that are entirely fabricated by Dr. Apon. When the underlying fabrication disappears, the contrast also disappears.

> [Next words in Dr. Apon's message:] We invited you to give a public 3rd Round Seminar talk on this issue in the Fall,

The reader will understand "this issue" to be the "new attack paper" against "cyclotomics". The reader is thus being told that NIST was asking me to give a talk specifically on the cyclotomic results; and that NIST was asking me to make the results available to the public. But both parts of this are fabrications by Dr. Apon.

NIST's actual invitation, by email dated 17 Aug 2020 16:36:05 +0000, was to give a talk "about the topic(s) of your choice in the security of lattice cryptography". The only specific lattice-security topic mentioned was "the state of the art of lattice reduction algorithms". Experts understand "lattice reduction algorithms" to include, e.g., LLL and BKZ, and to exclude, e.g., the algorithms featured in previous cyclotomic breaks. (I do have some work on reduction algorithms.)

In followup discussion, NIST said that "one area in particular that we would like to be as informed about as possible is the concrete security of lattice-based KEMs" against "lattice reduction", and that "another lattice topic we would like to hear about is an update on the status of any work towards algebraically cryptanalyzing structured lattices, e.g. closely approximating the shortest vector in (power-of-2) cyclotomic lattices".

So, no, it's not true that NIST invited me specifically to give a talk about cyclotomics. This was eventually on NIST's list of topics, but wasn't the only topic, and wasn't the first topic in the list, and wasn't the topic mentioned in NIST's first invitation message.

It's also not correct that the invitation was specifically to give a "public" talk. Having the talk be public was only an option mentioned "if you would prefer". I did state a preference for this option (email dated 3 Sep 2020 09:49:57 +0200):

> Procedurally, I now have some inkling of how much communication to and from NIST has been outside public view, and I am extremely uncomfortable with the lack of transparency, so I will certainly insist on having a public announcement and recording of any talk that I give to NIST.

Dr. Apon's fabrications regarding NIST's talk invitation are less important than Dr. Apon's central fabrications of timeline information, but they still tilt the overall picture.

> [Next words in Dr. Apon's message:] and you ended up giving a talk at the Seminar Series in January 15 of this year.

This is correct if "a talk" is taken out of context. However, the context makes the reader understand "a talk" as a "talk on this issue", i.e., a talk about the "new attack paper" against "cyclotomics"—and that's not true. Let's look at what actually happened.

In its initial invitation (email dated 17 Aug 2020 16:36:05 +0000), NIST had said that its talk series was about "important technical areas" and that "For many of these areas, we have 2 or 3 team members who understand the areas in greater depth, but the broader team does not have a good understanding of all these issues yet".

I wrote (email dated 3 Sep 2020 09:49:57 +0200): "So that I can better understand the context and objectives for this talk series, and what the audience is likely to know, can you please send me a list of the previous talks and other planned upcoming talks?" One of the examples NIST mentioned (email dated 3 Sep 2020 14:53:03 +0000) was an upcoming "introductory style talk on the basics of LLL/BKZ followed by their new results".

My assessment was that NIST was asking for remedial education. As noted above, given NIST's power over NISTPQC, people involved in NISTPQC really don't have the option of ignoring NIST's requests. But I was also extremely busy with other work items—for example, round-3 submissions filed in October 2020—so it took me some time to respond. I then commented (email dated 10 Nov 2020 13:51:36 +0100) that "it's informative, and disturbing, to learn that the audience doesn't already know the basics of LLL/BKZ".

NIST claimed (email dated 11 Nov 2020 17:29:56 +0000) to be "confident that the NIST PQC team members are familiar with e.g. LLL/BKZ (and at least reasonably familiar with all of the technical minutiae regarding more modern tweaks to lattice reduction algorithms and algebraic cryptanalysis of lattices over various number fields)".

I wrote (email dated 28 Nov 2020 13:28:34 +0100) "Can you please clarify what level of understanding you mean by 'reasonably familiar with all of the technical minutiae'? I'd expect this to mean that audience members would be able to fully define (e.g., implement) these algorithms without referring to notes, but I'm having trouble reconciling this with the September comment indicating NIST's interest in 'basics of LLL/BKZ' as part of someone's introductory talk." NIST never answered the question.

Based on the limited information available, I proposed (also in the 28 Nov 2020 13:28:34 +0100 email) a talk on "Valuations and S-units", with the following abstract:

> This talk reviews a standard infinite-dimensional number-theoretic lattice that simultaneously shows how large numbers are and how they factor. The ability to decode this lattice in some surprisingly large cases plays a critical role in a new wave of attacks against ideal-lattice problems. This talk will focus on defining the lattice, with many examples to illustrate.
>
> This is an introductory talk aimed at a broad audience. Prerequisites: mathematics education through a course in undergraduate abstract algebra (commutative rings and fields).

NIST said "this is perfect", and that's what the talk ended up being about.

> [Next words in Dr. Apon's message:] The talk presented a variety of algebraic and mathematical background that was quite interesting, but didn't suggest a clear attack vector.

The reader has been led to believe that NIST had invited me to give a talk about the "new attack paper" against "cyclotomics", and that I accepted—"but" then the talk didn't suggest an attack against cyclotomics.

It's not true that this is what NIST invited me to talk about. It's not true that this is what I agreed to talk about. The talk abstract that I proposed, and that NIST accepted, said that there's a standard number-theoretic lattice used in a new wave of attacks, and that this talk would focus on *defining* the lattice.

Does Dr. Apon inform the reader that attacks against cyclotomics were never within the agreed talk scope? No. Instead he deceives the reader into believing that I had been invited to give a talk on cyclotomic attacks, and had agreed to give a talk on cyclotomic attacks, "but" then failed to deliver.

[Next words in Dr. Apon's message:] At the time, I suggested that you finish the paper and submit the attack paper to this conference.

The recording shows that, in the Q&A session, I mentioned work in progress; Dr. Apon said "Do you have an ETA for this unreleased paper?"; and I said "Well, I'm one of six coauthors, so—it's one of these fun things where the results somehow keep getting better and better. The starting point was, okay, definitely smashed through this barrier, now let's see how much further we can go." Publications are, needless to say, driven by what makes sense scientifically.

Dr. Apon then sent email dated 15 Jan 2021 18:38:50 +0000 saying "I'm definitely looking forward to your unreleased paper making progress on the topic. As a reminder, the 3rd NIST PQC Standardization Conference is coming up; perhaps that is a good place to submit (and then also to a separate, academic conference with published proceedings as well)." So it's correct that he suggested this as a place to submit. So what? This wouldn't be remarkable without his timeline fabrications.

[Next words in Dr. Apon's message:] We didn't receive such a submission.

Dr. Apon has already told the reader that I had written to NIST 13–14 months ago to suggest that a new attack paper on cyclotomics was coming out in 2–3 months. Now Dr. Apon contrasts this with my not submitting any such paper to a NIST deadline a year after that.

It's correct that I didn't submit anything like this to the NIST conference.[5] But the entire contrast is against a "2–3 months" fabrication by Dr. Apon, not against something I ever said.

[Next words in Dr. Apon's message:] Given this background of no attack progress against cyclotomics since the beginning of the pandemic

What does the reader understand "progress" to mean here?

Often the word "progress" is comparing publications to earlier publications: this paper is making progress in the following way compared to that paper. Given the word "background", I'd expect the reader to understand Dr. Apon to be saying that, since the beginning of 2020, nobody has published any papers making progress on cyclotomic attacks. But then how does Dr. Apon explain https://eprint.iacr.org/2021/600, which was posted a month before Dr. Apon's text, and says in the abstract that it shows "that the decomposition group of a cyclotomic ring of arbitrary conductor may be utilised in order to significantly decrease the dimension of the ideal (or module) lattice required to solve a given instance of SVP"? That paper has certain limitations, but I don't see how Dr. Apon can claim that the paper isn't "progress".

It's possible that the reader instead understands Dr. Apon to be referring specifically to the project I'm involved in. The word "progress" in reference to one project generally refers to the steps inside that project, rather than comparing the project to previous work, so this type of reader understands Dr. Apon to be saying that this project hasn't made progress since the beginning of 2020. The case that Dr. Apon lays out for this relies entirely upon his fabricated timeline claims.

[Next words in Dr. Apon's message:] (after claiming at least an epsilon of progress would be coming in "a month or months" well over a year ago),

Given the context, the reader understands the quotation marks around "a month or

[5] Has NIST issued any statements regarding other people's supposed failure to submit something to NIST's conference? Not as far as I know.

months" to indicate that these words appeared in my email to NIST regarding this cyclotomic project. Dr. Apon is telling the reader that my email (1) claimed that something would already happen for this project in "a month or months" and (2) suggested that a paper was coming out for this project in 2–3 months.

But this "a month or months" quote is another fabrication by Dr. Apon. This is even worse than the earlier "coming out in 2-3 months" fabrication: not only does Dr. Apon attribute to the email something that the email simply did not say, but Dr. Apon inserts quotation marks so as to bolster the credibility of this fabrication.

> [Next words in Dr. Apon's message:] how would you characterize your progress in making a single epsilon of progress in attacking cyclotomic structures in lattice-based cryptography?

Starting from his fabricated timeline claims, Dr. Apon accuses me of deceiving NIST regarding the timeline, and then closes by publicly disputing that any progress has happened. He couches this dispute as a question. Since this isn't an honest question, it doesn't deserve an answer.

Venue and timing of Dr. Apon's accusation: disrupting a scheduled talk. NIST's Third PQC Standardization Conference took place online last week, 7–9 June 2021, as mentioned above. The conference charged a $25 registration fee but was open to the general public. NIST gave each round-3 submission team a 15-minute talk slot at the conference to present updates and field questions, and scheduled further talks regarding, e.g., performance comparisons. NIST also set up a chat system for the public to "submit questions and engage in group discussions" for the conference. The chat system and the video system each listed more than 300 people.

I gave the talk for one of the round-3 submissions, NTRU Prime, on behalf of a team of 10 people. Please note that I'm speaking for myself in this complaint.

NIST has full power over what happens in NISTPQC, and can send messages to submitters or to `pqc-forum` at any moment. Dr. Apon has sent 88 messages to `pqc-forum`. But **Dr. Apon chose to issue his accusation during the NTRU Prime talk**.

The talk was scheduled to run from 15:25 to 15:40 D.C. time. However, this was the fourth talk in the session, and the session was running late. My best reconstruction based on the data available at this point is that the talk ran from 15:34 to 15:49 D.C. time. Dr. Apon published his accusation on the chat system at 15:38 D.C. time.

As noted above, Dr. Apon introduced his accusation as "a quick, gentle question", followed by lengthy text. The typical reader would be captured by the clickbait; read further; see the accusation of misbehavior in Dr. Apon's first sentence after the introduction; find this interesting; continue reading, finding further text amplifying the accusation of misbehavior; and, eventually, find a "question" at the very end of Dr. Apon's text. Any such reader would have been

- distracted from listening to the NTRU Prime talk, and
- immediately prejudiced against the NTRU Prime talk, given these accusations against the speaker.

Note that NIST relies heavily on public analysis of submissions—for example, public work to make submissions perform well in various environments. The talk was not just for NIST but for the entire audience, explaining reasons to be interested in NTRU Prime. It's easy to predict that Dr. Apon's disruption will end up taking some public analysis away from NTRU Prime.

Presumably Dr. Apon will respond that this talk disruption was justified. Dr. Apon knew that my talk was going to mention the history of attack advances *against cyclotomics* and the possibility of further attack advances *against cyclotomics*; Dr. Apon's accusation is specifically that I had deceived NIST regarding the timing of an announcement *regarding cyclotomics*; Dr. Apon thought it was important for people listening to the talk to know this, and to distrust the talk accordingly.

However, *sometimes* accusations are false (as illustrated by Dr. Apon's accusation), which is why *in all cases* it's important to follow proper dispute-resolution procedures, including normal due-process safeguards. Did Dr. Apon follow these procedures before broadcasting his accusation at a conference attended by more than 300 people? No, he didn't. Did he tell Dr. Moody what he was planning to do? According to Dr. Moody, no, he didn't. Perhaps Dr. Apon will try to claim that his fabrications regarding my email (and regarding NIST's talk invitation and so on) were merely the result of an amazing series of memory failures rather than dishonesty; but did he *check* the email before issuing his accusation? No, he didn't.

Furthermore, anyone who looks at the talk slides or listens to the talk video (I've made the slides and video available at `https://cr.yp.to/talks.html#2021.06.08`) can see that the talk was covering many points other than cyclotomics. It appears that Dr. Apon posted his 216-word "Regarding cyclotomic-based attacks" text at the first moment cyclotomics were mentioned—but, before this, the talk had already covered

- the fact that public security reviewers are overloaded;
- a broad overview of recent advances in lattice attacks;
- the failure of "provable security" to stop small lattice systems from being broken;
- performance requirements sometimes forcing the use of small lattice systems;
- NTRU Prime's goal of, and success in, reducing the attack surface for small lattice systems;
- decryption failures as an example of a NISTPQC attack tool that NTRU Prime had eliminated from the outset; and
- the advantages of proactively reducing attack surface rather than merely reacting to breaks.

Cyclotomics then appeared as another example of a NISTPQC attack tool that NTRU Prime had eliminated from the outset—but, even if Dr. Apon objected to this example, how could this objection possibly have been so important as to justify disrupting a talk that was obviously broader than this? Why not allow the talk to finish, and calmly follow up with objections afterwards?

Subsequent events. The conference organizers at NIST had offered the option of sending a video in advance. I had taken this option, because of concerns regarding time zones and regarding difficulties connecting to NIST's video system. Like typical audience members, I was watching the chat system for questions while the video played.

I was, to put it mildly, surprised seeing Dr. Apon's "question" on the chat system. I saw that the message was starting by deceiving readers regarding NIST's request for input, so I replied regarding that:

> djb 9:39 PM please don't misrepresent the history. nist specifically asked to be informed regarding ongoing projects, and i answered.

Dr. Apon replied, repeating his "question" to avoid commenting on his misrepresentation of the history—a classic example of an ad-hominem attack:

> Daniel Apon 9:39 PM So, no progress?

Reading further in Dr. Apon's text, I found one outright fabrication after another, and responded accordingly:

> djb 9:44 PM i'll post an apon fact check to pqc-forum in due time. in the meantime, happy to answer honest questions.
>
> [emoji reactions: "eyes" from Daniel Apon; "hushed" from Yuji Suga]

The damage to the talk was already done. The only further questions at the conference regarding the talk were from NIST employees:

> angela.robinson 9:51 PM I am curious to know if you are planning a follow-up talk to your talk given at the public NIST PQC seminar series. @djb
>
> [emoji reaction: "+1" from Daniel Apon]
>
> djb 9:54 PM @angela.robinson Thanks for your question! As I mentioned in the talk, there's a second part coming up. This is scheduled for August.
>
> angela.robinson [reply in thread] Great! Sorry if I missed that part. The pace was a bit fast, but I look forward to it.
>
> Daniel Apon [reply in thread] Looking forward to it!
>
> djb 9:56 PM To clarify since there are multiple talks at issue: I mentioned in the January talk that a second talk was coming up on S-unit attacks; that talk is now scheduled for August; I didn't mention that talk in my talk today.
>
> [emoji reactions: "+1" from Angela Robinson and Daniel Apon]
>
> Daniel Apon [reply in thread] Where will the talk be?
>
> Daniel Smith-Tone 10:04 PM @djb Did the parameters for sntrup4591761 and ntrulpr4591761 change between round 1 and round 2? I am asking because in round 1 the specification document claims NIST security level 5, but then level 3 is claimed in the round 2 spec and afterwards.
>
> djb 10:14 PM @Daniel Smith-Tone I believe the submission document addresses everything, but let me try to answer your question, speaking for myself. There's no change in the parameter set. There have been worrisome advances in attacks against all small-lattice submissions. There are also massive ambiguities in NIST's definitions of security levels, as illustrated by NIST's SHA3-256 security evaluation jumping from 2^80 to 2^146 in 2016. Some ambiguities in NIST's definitions of security levels were partially resolved by NIST statements in round 1, so the NTRU Prime assignments were revised accordingly. Many problematic ambiguities remain; see generally `https://cr.yp.to/papers.html#categories`.
>
> Daniel Smith-Tone 10:15 PM I think that the submission document does address this, but I thought I would ask directly since the source is available. I also agree with some of what you just said.
>
> djb 10:15 PM Section 5.4 of that document identifies various specific definitional questions that NIST still hasn't answered.
>
> Daniel Smith-Tone 10:16 PM I'm reading that right now.
>
> djb 10:17 PM Please let me know if anything is unclear.

Meanwhile I wrote to Dr. Apon (email dated 8 Jun 2021 21:53:21 +0200) as follows:

> I'm writing to request a retraction of the "question" that you issued during my talk. I expect the retraction to include a clear and specific acknowledgment that you

> fabricated the “coming out in 2-3 months” part, and that this part plays a critical role in your “question”. I’ll give you 1 week before escalating.

Later I wrote to Dr. Apon (email dated 8 Jun 2021 23:54:44 +0200) as follows:

> Appendix: I also expect the retraction to include the same clear and specific acknowledgment regarding your fabrication of the “month or months” part, which plays the same role in your “question”.

I wrote to Dr. Moody, cc’ing Dr. Apon and the NTRU Prime team, as follows (email dated 9 Jun 2021 00:03:20 +0200):

> As you know, each round-3 submission team was given a 15-minute slot at the ongoing NIST conference to present updates for, and field questions from, an online audience of about 300 people, of course including NIST.
>
> For NTRU Prime in particular, I gave the talk. I presume you saw that, early in the talk, Dr. Apon posted a “question” to the Slack channel, a “question” that typical audience members would understand as accusing me of misbehavior. A copy of his accusation appears below.
>
> I will, needless to say, take appropriate action to respond to these accusations. I’ve already contacted Dr. Apon regarding the substance of the accusations, and have offered him a week before I escalate.
>
> However, in the meantime, I believe it’s appropriate to immediately address the following procedural question to you, since Dr. Apon is, if I understand correctly, a NIST PQC team member under your supervision. If I’ve misunderstood the management structure within NIST and should contact Dr. Chen instead, please let me know.
>
> Dr. Apon is a member of the NIST PQC team. The NIST PQC team has full power over what happens in this competition, and can ask questions on pqc-forum at any moment. However, Dr. Apon chose to issue accusations *during the NTRU Prime talk*. Given the length, timing, and nature of his text, it seems reasonably clear that he prepared the text in advance and chose the timing to maximally disrupt the talk. Obviously the video kept playing, but many people
>
> * watch the Slack channel for the occasional questions while listening to talks,
> * would have been distracted from listening because of the obviously interesting nature of Dr. Apon’s text, and
> * would then have been prejudiced against the talk because of the accusations communicated by the text.
>
> So here’s the question for you: Did you know he was going to do this?
>
> Since what happened here is an issue for the NTRU Prime team as a whole, I’m cc’ing the team.

Dr. Apon replied as follows (email dated 9 Jun 2021 00:10:48 +0000):

> My question was asked honestly and in earnest (that is, as a person sincere and serious in behavior and convictions).
>
> I think your question is bested addressed to my supervisors. I’ve included Lily Chen and Matt Scholl on the CC list. You’re welcome to discuss with them.

Dr. Moody replied as follows (email dated 9 Jun 2021 12:51:39 +0000):

> Thank you for your message and sharing your concern with me. As a quick re-

> sponse to your question - no, I did not know in advance that Daniel Apon would ask you a question on slack during the NTRUprime update. Our PQC team members can certainly post questions without needing to ask me. I actually had to skip the last 30 minutes of the day (my son was running his last race of his high school career at a track meet), so I wasn't following live. I didn't check back into work things until this morning.
>
> I am sorry if you felt his question disrupted your talk. I'll talk to Daniel, but from his response to you he says he wasn't intending to disrupt, but wanted to ask you an honest question.

Concluding remarks. I've reviewed this complaint, am confident in its accuracy, and see many reasons to have it online as soon as possible. Am I simply posting it?

No. I'm sending it to a limited audience: Dr. Apon (the subject of this complaint), Dr. Moody (in charge of NISTPQC), Dr. Chen (the supervisor that this complaint is addressed to), Dr. Matthew Scholl (Dr. Apon already brought Dr. Scholl into the discussion), and the NTRU Prime team (directly damaged by Dr. Apon's talk disruption, beyond the damage caused by the content of Dr. Apon's accusation).

Why am I not posting the complaint immediately? Because that wouldn't be the proper procedure. I am accusing Dr. Apon of misconduct, and he is entitled to an opportunity to defend himself. I can't imagine how he can justify his ludicrously unprofessional, clearly dishonest actions—but that's not the point. Sometimes people *can* successfully defend themselves, so civilized society sets up procedures that recognize this and that, as noted above, are to be followed in *all* cases.

Part of what I'm complaining about, obviously, is that Dr. Apon didn't follow such procedures: on the contrary, he simply went ahead and *published* his accusation. Given that the accusation is public, I'm certainly entitled to defend myself in public (while, for comparison, Dr. Apon has no such excuse for his procedural violations), so I could easily justify posting this complaint today. Delaying this posting means that Dr. Apon's fabrications will have more time to spread and will cause more damage. But delaying this posting—for a limited time—is still the right thing to do.

One week is ample time for NIST management to compare all relevant email to Dr. Apon's fabrications and to take appropriate action. Given the nature of Dr. Apon's action, I expect this matter to be given high priority. I will, of course, give due consideration to anything that Dr. Apon says now in his defense, and I will give due consideration to any explanation of why more time is requested.

From: 'Moody, Dustin (Fed)' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Thursday, September 16, 2021 7:49 AM
To: D. J. Bernstein; pqc-forum
Subject: [pqc-forum] Re: ROUND 3 OFFICIAL COMMENT: NTRU Prime

Dan,

That there was an impolite exchange on a Slack thread does not mean the PQC standardization process is unfair or biased in any way. We all have individual perspectives, as researchers and scientists, and it is inevitable that some disagreements and misunderstandings will occur during this process. However, we work as a team to ensure that the final outcomes of this process are rigorous and fair. We take our responsibilities very seriously, and we will continue to be fair and impartial as we evaluate the different algorithms. We make our decisions scientifically based upon the technical results we see published or presented. The community is welcome to provide feedback if they feel this is not the case.

Also, as a reminder, we would like to keep the pqc-forum primarily for technical discussions. We will follow up with you directly on other issues.

Dustin
The NIST PQC team

From: D. J. Bernstein
Sent: Wednesday, September 8, 2021 10:16 PM
To: pqc-forum
Cc: pqc-comments
Subject: ROUND 3 OFFICIAL COMMENT: NTRU Prime

Each round-3 submission team was given a 15-minute slot at the NIST conference three months ago to present updates for, and field questions from, an online audience of about 300 people, of course including NIST.

During the NTRU Prime talk, Dr. Apon posted the text quoted below to the Slack channel for the conference, publicly accusing me of professional misconduct. Specifically, he accused me of initiating private contact with NIST so as to provide false information to NIST regarding the timing of an upcoming announcement relevant to NIST's ongoing decisions. However:

* The contact was requested by NIST.

* The false information that Dr. Apon attributes to me is a fabrication by Dr. Apon.

From: pqc-forum@list.nist.gov on behalf of Christopher J Peikert <cpeikert@alum.mit.edu>
Sent: Thursday, September 30, 2021 10:30 AM
To: pqc-forum
Subject: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

Summary of this comment:

1. The NTRU Prime FAQ starts with an objectively false factual claim about competing submissions Kyber and SABER.
2. This false claim is central to the FAQ's misleading attempt to suggest that these systems infringe on a patent.
3. Requests to the NTRU Prime team to remove the false claim and insinuation were refused.
4. I therefore believe that this is not an honest mistake, but a deliberate attempt to smear competing proposals with false disparaging claims and FUD.
5. I request that NIST consider what to do about patterns of behavior like this.

The first answer of the NTRU Prime FAQ, which appears on the official project website [link] and was repeated by Dan Bernstein on the pqc-form on 11 December 2020 [link], says this (emphasis added):

"There are known patent threats against ... Kyber, SABER, and NTRU LPRime (ntrulpr). **These proposals use ... a 2x ciphertext-compression mechanism** that **appears to be covered** by U.S. patent 9246675 expiring 2033."

(The ellipses replace another claim of threat from a different patent, which is outside the scope of this message, but was also shown to be severely flawed; see, e.g., [link, link].)

As a matter of objective fact, the "2x" claim is false. While Kyber and SABER do perform some mild ciphertext compression, they do not, and could not, come close to 2x with the mechanism they use.

(I pointed out this false "2x" claim on the pqc-forum on 11 December 2020 [link], and again on 21 May 2021 [link], and in private correspondence with the NTRU Prime team, with an explicit request to correct it, but the team refused.)

Why does this matter?

The false "2x" claim is central to the FAQ's attempt to tie Kyber and SABER to the cited patent. Specifically, the "appears to be covered" claim implicitly conflates the patented mechanism, which does provide (near-)2x compression, with the unpatented prior-art method that Kyber/SABER use.

Kyber/SABER's compression mechanism is, informally: "drop some low bits of certain integers, keeping the several high bits needed for correct decryption." This method appears in at least four well known works of prior art to the cited patent, some of which are cited in every version of the Kyber submission. For details, see the last part of my pqc-forum message from 21 May 2021 [link].

The patent describes a *different* compression mechanism whose main benefit is that it can provide near-2x in certain contexts, by keeping just a single bit of certain integers. Kyber/SABER do not use this method, and the patent does not claim the above prior-art method that they do use. A detailed explanation of the prior art and the differences between the methods is given in my pqc-forum message from 22 May 2021 [link].

In private correspondence with the NTRU Prime team, based on the above reasoning I stated that **the FAQ's "appears to be covered" claim is highly misleading**, and requested that it be removed. The team refused.

The above summarizes, for the record, the history regarding the facts and analysis. The rest of this message contains my conclusions about the situation, and a discussion of how to proceed from here.

I believe that the FAQ entry is a deliberate attempt to smear competing proposals with false disparaging claims and FUD. Of course, the false "2x" claim could originally have been an unintentional error---albeit a sloppy one, showing unfamiliarity with basic properties of the schemes.

However, the team's refusal to fix even this elementary factual error leads me to conclude that the claim has been made intentionally to deceive, i.e., to conflate the unpatented prior art with the patent's near-2x method, and to misleadingly suggest that Kyber/SABER infringe on the patent. Without "2x," there's no link to the patent, and the FAQ entry falls apart (along with subsequent entries that are premised on it).

What next?

I hope the above material sets the record straight. But this example raises the broader issue of NIST PQC participants who exhibit a pattern of the following behavior:

1. Falsely disparage other submissions and/or the process itself.
2. Receive corrections showing these claims to be factually false or otherwise meritless.
3. Make no withdrawal of the false claims. Even worse, give no acknowledgment of the corrections. Even worse than that, persist in spreading the false claims.

(Some other examples of this pattern appear at the end of this message.)

This kind of behavior is outside the bounds of fair play. It sows confusion among non-experts who may only be able to see a "controversy," and it badly wastes the community's time that could be better spent on more productive matters. (Brandolini's law estimates the cost at 10x, but I think that's too low in this context.)

To be absolutely clear: I am not talking about honest mistakes or misunderstandings that are acknowledged and corrected. Indeed, this describes the vast majority of situations in the NIST PQC process, in which submitters and other participants have resolved matters without difficulty.

Procedurally, I think NIST should seriously consider this issue. I can think of a few options for how it could respond, such as:

1. Take no official action. Let people say whatever they want to, and hope that other (unspecified) mechanisms address such behavior. This has the big disadvantage that it does not offer any clarity to non-experts and the broader community.
2. Make an official statement on its findings of the relevant facts, and perhaps its analysis of the consequences. This has the advantage of offering clarity to the community.
3. Do 2, and also penalize submissions/submitters who show a pattern of this kind of behavior, perhaps after a warning and a failure to remedy matters. This has the additional advantage of providing a disincentive to wasting the community's time with FUD and nonsense.

As mentioned above, here are two more examples fitting the pattern of false disparagement, followed by debunking, with no withdrawal or even acknowledgment:

1. The false accusation that round-3 Kyber "switched from Core-SVP to a modified metric," which was conclusively shown [link] to be based on nothing but the accuser's severe misunderstanding (or worse, deliberate mischaracterization) of what Core-SVP means.

2. The striking accusation that "NIST started trying, with considerable success, to delay and deter public analysis of the patent threats" [link]. A follow-up message [link] requested evidence to support this accusation---which was never provided---and showed prior statements from NIST *encouraging* comments about patent issues.

Sincerely yours in cryptography,
Chris
--

From:	pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent:	Thursday, October 7, 2021 6:20 AM
To:	pqc-comments
Cc:	pqc-forum
Subject:	Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments:	signature.asc

Executive summary: The "OFFICIAL COMMENT" that I'm replying to is incorrect regarding every topic of dispute. The errors range from (1) obvious to (2) more subtle but already discussed in detail on pqc-forum.
The "OFFICIAL COMMENT" also misrepresents its disputed arguments as undisputed, which is inappropriate.

Details follow, beginning with a review of ciphertext sizes and continuing with replies to the "OFFICIAL COMMENT".

Regarding 2x ciphertext compression, the numbers speak for themselves:

* 1568-byte ciphertext, including suboptimal encoding and CCA protection: Kyber-1024. This uses modulus q=3329.

* 2996-byte ciphertext, with optimal encoding, no CCA protection: LPR with the ring (\Z/3329)[x]/(x^1024+1). An LPR ciphertext has 2 ring elements, and ceil(2*1024*log(3329)/log(256)) = 2996.

1568 is 52% of 2996. Here's another example:

* 1472-byte ciphertext, with optimal encoding, including CCA protection: FireSABER. This uses modulus q=8192.

* 3328-byte ciphertext, with optimal encoding, no CCA protection: LPR with the ring (\Z/8192)[x]/(x^1024+1).

1472 is 44% of 3328. Here's another example:

* 2208 bytes, including suboptimal encoding and CCA protection: submitted NewHope-1024. This uses modulus q=12289.

* 3478 bytes, with optimal encoding, no CCA protection: LPR ciphertexts for the ring (\Z/12289)[x]/(x^1024+1).

2208 is 63% of 3478. Here's another example:

* 2048 bytes, including suboptimal encoding, no CCA protection: original NewHope-1024. This uses modulus q=12289.

* 3478 bytes, with optimal encoding, no CCA protection: LPR ciphertexts for the ring (\Z/12889)[x]/(x^1024+1).

2048 is 59% of 3478. Original NewHope, in turn, was based on BCNS, which was based on https://eprint.iacr.org/2014/070, which highlighted smaller ciphertexts than LPR as something new---

As compared with the previous most efficient ring-LWE
cryptosystems and KEMs, the new reconciliation mechanism reduces the
ciphertext length by nearly a factor of two

---and correctly stated that an LPR ciphertext has "two R_q elements".
Retroactive efforts at obfuscation do not have the power to stop a patent court from reaching clarity regarding the ciphertext sizes.

The quoted statement from https://eprint.iacr.org/2014/070 also has an important error that would be identified in a patent case. The statement isn't simply a claim to be 2x better than LPR, but rather a claim to be 2x better than the "previous most efficient ring-LWE cryptosystems and KEMs"---which isn't true. Ding's paper https://eprint.iacr.org/2012/688 was two years earlier, and was already 2x smaller than LPR.

The continued failure of https://eprint.iacr.org/2014/070 to give appropriate credit to Ding is in violation of basic ethics rules (see, e.g., https://ori.hhs.gov/plagiarism-ideas) and, as a historical matter, led directly to

* BCNS at first being unaware of Ding's contribution,
* NewHope at first being unaware of Ding's contribution, and
* Google at first being unaware of Ding's contribution.

Google rolled out a big experiment with NewHope in July 2016, saying that it wanted "to help ensure our users' data will remain secure long into the future" and that it would end the experiment "within two years, hopefully by replacing it with something better". Google suddenly ended the experiment a few _months_ later, after Ding reportedly contacted Google about his patent. Google then waited _three years_ before trying a new post-quantum experiment---and didn't select an LPR-type system.

Late-2016 awareness of Ding's patent prompted various efforts to build LPR-type systems that work around the patent without regressing to the original LPR ciphertext sizes. The most interesting idea is to build LPR-type systems without "reconciliation" and thus avoid Ding's patent.
This sounds great if it works. Unfortunately, a careful examination shows that this dividing line is ill-defined---which is a huge problem in a patent case, since the procedures used in patent courts force everything to be defined. See the analysis in my pqc-forum message dated
1 Jan 2021 13:19:26 +0100.

Christopher J Peikert writes:
> Summary of this comment:
>   1. The NTRU Prime FAQ starts with an objectively false factual claim
>   about competing submissions Kyber and SABER.

No, the FAQ (https://ntruprime.cr.yp.to/faq.html) is correct as stated.

>   2. This false claim

Again, the FAQ is correct as stated.

> is central to the FAQ's misleading attempt to
>   suggest that these systems infringe on a patent.

The FAQ correctly warns people about patent _threats_. This is an important public service. Unlike some overconfident commentators, the FAQ also notes the uncertainties here:

Perhaps the appeal regarding 9094189, and subsequent litigation
regarding both patents, will succeed in eliminating these patents or
limiting their coverage. However, today it is far from clear that
"Product NTRU"/"Ring-LWE"/"LPR" systems will be free to use before
2033.

The NISTPQC call for submissions says that it is "critical that this process leads to cryptographic standards that can be freely implemented in security technologies and products". NIST's analyses of the patent threats should have been online years ago for public review.

> 3. Requests to the NTRU Prime team to remove the false claim and
> insinuation were refused.

The FAQ is correct as stated. The recent email messages that Dr. Peikert characterizes as "requests" were inappropriate and triggered a complaint from the NTRU Prime team to NIST dated 28 Sep 2021 20:39:53 +0300, also sent to Dr. Peikert.

> 4. I therefore believe that this is not an honest mistake, but a
> deliberate attempt to smear competing proposals with false disparaging
> claims and FUD.

Once again, the FAQ is correct as stated.

> 5. I request that NIST consider what to do about patterns of behavior
> like this.

See below.

> The first answer of the NTRU Prime FAQ, which appears on the official
> project website [link <https://ntruprime.cr.yp.to/faq.html>] and was
> repeated by Dan Bernstein on the pqc-form on 11 December 2020 [link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/NSe0wAzKJtA/m
> /DSIMaEOcCAAJ>],
> says this (emphasis added):
> "There are known patent threats against ... Kyber, SABER, and NTRU
> LPRime (ntrulpr). *These proposals use ... a 2x ciphertext-compression
> mechanism* that *appears to be covered* by U.S. patent 9246675 expiring
2033."
> (The ellipses replace another claim of threat from a different patent,
That patent, U.S. patent 9094189, is another patent threatening the LPR-type systems. This is also why NIST tried to buy out the patent.

> which is outside the scope of this message, but was also shown to be
> severely flawed; see, e.g., [link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/NSe0wAzKJtA/m
> /SDaWm_qtCAAJ>,
> link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/NSe0wAzKJtA/m
> /DxGrdfWHAgAJ>
>].)

Nope. See, e.g., my pqc-forum email dated 17 May 2021 12:09:27 +0200.

> *As a matter of objective fact, the "2x" claim is false.*

The 2x claim is correct. See numbers above.

> While Kyber and SABER do perform some mild ciphertext compression,
> they do not, and could not, come close to 2x with the mechanism they
> use.

They certainly do come close to 2x. See numbers above. Also, the notion that the exact magnitude is important was addressed in my pqc-forum message dated 1 Jan 2021 13:19:26 +0100: "It's normal in patent cases for defendants to try to avoid a patented efficiency improvement by interpolating between the prior art and the efficiency improvement, and it's normal for the patentee to win."

> (I pointed out this false "2x" claim on the pqc-forum on 11 December
> 2020 [link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/NSe0wAzKJtA/m
> /MMJdZOq1CAAJ>],
> and again on 21 May 2021 [link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/nbIZhtICKWU/m
> /SOk7xQBsBQAJ>], and in private correspondence with the NTRU Prime
> team, with an explicit request to correct it, but the team refused.)
> Why does this matter?
> *The false "2x" claim is central to the FAQ's attempt to tie Kyber and
> SABER to the cited patent.*

The 2x claim is correct. See numbers above.

> Specifically, the "appears to be covered" claim implicitly conflates
> the patented mechanism, which does provide (near-)2x compression, with

No. Everything is explicit in various pqc-forum messages, notably my email dated 1 Jan 2021 13:19:26 +0100, which goes through the relevant systems in detail and certainly doesn't conflate anything.

> the unpatented prior-art method that Kyber/SABER use.

Now _this_ is an example of conflation, where a claimed _analogy_ between Kyber and prior-art systems is being misstated as an _equality_.
That's not how patent cases work.

> Kyber/SABER's compression mechanism is, informally: "drop some low
> bits of certain integers, keeping the several high bits needed for
> correct decryption."

The "mechanism" concept used here is divorced from patent law.

> This method appears in at least four well known works of prior art to
> the cited patent, some of which are cited in every version of the
> Kyber submission. For details, see the last part of my pqc-forum
> message from 21 May 2021 [link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/nbIZhtICKWU/m
> /SqvzrtVOBQAJ>

>].

How many of those publications presented compressed LPR? Zero. A lawyer would have to try to turn this into an _obviousness_ argument, saying that compressed LPR was _obvious_ from the prior art, but then the opposing lawyer pulls out a 2014 paper from an expert saying

   One of our main technical innovations ... reduces the ciphertext
   length of prior (already compact) encryption schemes nearly twofold
   ... As compared with the previous most efficient ring-LWE
   cryptosystems and KEMs, the new reconciliation mechanism reduces the
   ciphertext length by nearly a factor of two, because it replaces one
   of the ciphertext’s two R_q elements with an R_2 element

which clearly states that no previous work had compressed the ciphertext below "two R_q elements". If an expert in 2014 was claiming this as new, the result of an "innovation", how can it have been obvious in 2012?

> The patent describes a *different* compression mechanism whose main
> benefit is that it can provide near-2x in certain contexts, by keeping
> just a single bit of certain integers. Kyber/SABER do not use this
> method, and the patent does not claim the above prior-art method that
> they do use. A detailed explanation of the prior art and the
> differences between the methods is given in my pqc-forum message from
> 22 May 2021 [link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/NSe0wAzKJtA/m
> /uYJ3W-_RBQAJ>
>].

See above regarding (1) "mechanism", (2) the notion that exact magnitude is important here, and (3) the ill-defined claimed dividing line between Kyber/SABER and what's patented.

> In private correspondence with the NTRU Prime team, based on the above
> reasoning I stated that *the FAQ's "appears to be covered" claim is
> highly misleading*, and requested that it be removed. The team refused.

See above regarding this "request".

> The above summarizes, for the record, the history regarding the facts
> and analysis.

"For the record"? The cited messages are already on the record, except for the recent "request".

> The rest of this message contains my conclusions about the situation,
> and a discussion of how to proceed from here.
> *I believe that the FAQ entry is a deliberate attempt to smear
> competing proposals with false disparaging claims and FUD.* Of course, the false "2x"
> claim could originally have been an unintentional error---albeit a
> sloppy one, showing unfamiliarity with basic properties of the schemes.

Let's see here:

   * On one side, compression of 2996 bytes to 1568 bytes, 52%, is being
     described as "some mild ciphertext compression" but nothing "close

to 2x".

* On the other side, compression of 2996 bytes to 1568 bytes, 52%, is being described as "2x" and "twofold".

I don't think any commentary is needed here for the reader to see which side is correct. I'll skip commenting on the endless personal attacks.

> However, the team's refusal to fix even this elementary factual error
> leads me to conclude that the claim has been made intentionally to
> deceive, i.e., to conflate the unpatented prior art with the patent's
> near-2x method, and to misleadingly suggest that Kyber/SABER infringe
> on the patent. Without "2x," there's no link to the patent, and the
> FAQ entry falls apart (along with subsequent entries that are premised on it).

This whole "refusal to fix" narrative, exploration of consequences, and so on tries to force the reader to imagine that there's something to fix in the first place. There isn't.

> *What next?*
> I hope the above material sets the record straight.

It doesn't. This "OFFICIAL COMMENT" is wasting everyone's time by repeating previous errors with marginally different wording.

> But this example raises the broader issue of NIST PQC participants who
> exhibit a pattern of the following behavior:
>   1. Falsely disparage other submissions and/or the process itself.
>   2. Receive corrections showing these claims to be factually false or
>   otherwise meritless.
>   3. Make no withdrawal of the false claims. Even worse, give no
>   acknowledgment of the corrections. Even worse than that, persist in
>   spreading the false claims.

Anyone who looks at the detailed analysis in my message dated 1 Jan 2021
13:19:26 +0100 can see that the above is not a defensible description of the history. It's disturbing to see this "OFFICIAL COMMENT" trying to add undeserved weight to its incorrect position by spreading outright misinformation regarding the status of the discussion.

The loaded word "disparagement" above seems likely to trigger emotional reactions, so let's take a moment to consider how the Frodo submission

* includes a worst-case-to-average-case reduction in its list of "reductions supporting the security of FrodoKEM" and

* highlights "several lattice-based proposals that lacked such reductions, and turned out to be insecure".

This is the obvious source of, e.g., NIST IR 8309 claiming that NTRU "lacks a formal worst-case-to-average-case reduction" and not saying the same regarding Frodo. However, the claim that Frodo has an advantage here---that it has reductions that, e.g., NTRU does not have---is wrong.
See, e.g.,

* my pqc-forum email dated 21 Apr 2018 22:15:53 -0000,
* my pqc-forum email dated 22 Apr 2018 21:04:33 -0000 (which described the security-failure mode that we then saw with MQDSS),
* my pqc-forum email dated 23 Apr 2018 15:42:22 -0000, and
* Section 9 of https://cr.yp.to/papers.html#latticeproofs.

It's now 3 years later, and the Frodo submission continues to spread the same claim. Note that the negative effect of the claim upon NTRU doesn't rely upon the Frodo submission identifying NTRU as an example.

> (Some other examples of this pattern appear at the end of this
> message.)

See below.

> This kind of behavior is outside the bounds of fair play. It sows
> confusion among non-experts who may only be able to see a
> "controversy," and it badly wastes the community's time that could be
> better spent on more productive matters. (Brandolini's law estimates
> the cost at 10x, but I think that's too low in this context.)

The above paragraph nicely captures why this time-wasting "OFFICIAL COMMENT" should not have been filed in the first place.

> To be absolutely clear: I am not talking about honest mistakes or
> misunderstandings that are acknowledged and corrected. Indeed, this
> describes the vast majority of situations in the NIST PQC process, in
> which submitters and other participants have resolved matters without difficulty.

This claim is hard to evaluate unless "without difficulty" is clarified.
As an example, when Frodo falsely claimed a "Theorem 5.1" assuming merely OW-CPA rather than IND-CPA, made a synchronized series of changes in support of this claim, didn't admit the error for a month and a half after it was pointed out, and then tried to downplay the error as a "typo", would this qualify as resolution "without difficulty"? See the discussion that started with my 24 May 2019 08:33:24 -0000 message, and see Section 6 of https://cr.yp.to/papers.html#latticeproofs.

> Procedurally, I think NIST should seriously consider this issue. I can
> think of a few options for how it could respond, such as:
>   1. Take no official action. Let people say whatever they want to, and
>   hope that other (unspecified) mechanisms address such behavior. This has
>   the big disadvantage that it does not offer any clarity to non-experts and
>   the broader community.
>   2. Make an official statement on its findings of the relevant facts, and
>   perhaps its analysis of the consequences. This has the advantage of
>   offering clarity to the community.
>   3. Do 2, and also penalize submissions/submitters who show a pattern of
>   this kind of behavior, perhaps after a warning and a failure to remedy
>   matters. This has the additional advantage of providing a disincentive to
>   wasting the community's time with FUD and nonsense.

Seems unnecessary to comment on this.

> As mentioned above, here are two more examples fitting the pattern of

> false disparagement, followed by debunking, with no withdrawal or even
> acknowledgment:
> 1. The false accusation that round-3 Kyber "switched from Core-SVP to a
> modified metric," which was conclusively shown [link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/NSe0wAzKJtA/m/JmAHc1GqCAAJ>]
> to be based on nothing but the accuser's severe misunderstanding (or worse,
> deliberate mischaracterization) of what Core-SVP means.

The switch of metric is documented in detail in my messages dated 4 Dec 2020 18:06:07 +0100, 17 Dec 2020 16:20:01 +0100, and 2 Jan 2021 18:24:27 +0100. See also the ten questions in my message dated 4 Jan 2021 14:39:45 +0100. I'll again skip commenting on the personal attacks.

> 2. The striking accusation that "NIST started trying, with considerable
> success, to delay and deter public analysis of the patent threats" [link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/nbIZhtICKWU/m/ML7aYY71AgAJ>].
> A follow-up message [link
> <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/nbIZhtICKWU/m/SqvzrtVOBQAJ>]
> requested evidence to support this accusation---which was never
> provided---and showed prior statements from NIST *encouraging* comments
> about patent issues.

Let's review. This "OFFICIAL COMMENT" is extracting the ending of the following note:

NIST posted the IP statements three years ago. ... However---even though the call for proposals had described free usability as "critical"---NIST started trying, with considerable success, to delay and deter public analysis of the patent threats.

The "OFFICIAL COMMENT" is characterizing this note as "false disparagement" of the "process". It's then pointing to email dated 21 May 2021 00:35:37 -0400 as "debunking" this "disparagement", where the "debunking" consists of showing "prior statements from NIST *encouraging* comments about patent issues."

Now let's compare this to the facts:

* NIST email dated 9 Jan 2018 18:18:08 +0000 says "We would also appreciate any comments from the community of course." This is quoted in the email dated 21 May 2021 00:35:37 -0400, which this "OFFICIAL COMMENT" characterizes as a "debunking" showing "prior statements from NIST *encouraging* comments about patent issues".

* NIST email four months later, dated 8 May 2018 13:35:17 +0000, announced that NIST had posted the IP statements. The note quoted above refers to this posting and then says "NIST started trying, with considerable success, to delay and deter public analysis of the patent threats".

Even if we imagine the January 2018 message as requesting public analysis of the patent threats, how is this supposed to be "debunking" a note referring to NIST's May 2018 posting of the IP statements and saying that NIST _started_ trying to delay/deter analysis? Note the word "started"; words have meanings.

Structurally, the timeline directly disproves the "debunking" narrative.
Of course, the "OFFICIAL COMMENT" omits "posted the IP statements", hides the timeline from the reader, and indefensibly characterizes this irrelevant January 2018 reference as "debunking" a note about what happened later. When this error is stripped away, we're left with "requested evidence to support this accusation---which was never provided", which obviously also doesn't qualify as "debunking". The reader is being fed outright misinformation regarding the status of the discussion; again, this is not appropriate.

Any reader who looks at the same NIST email dated 8 May 2018 13:35:17
+0000, the one where NIST announced the signed IP statements, sees that
NIST then said "For the 1^ st round, all submissions should be evaluated and analyzed on their technical merits". That's a delay all the way to 2019---for the only item labeled as "critical" in the call for submissions!

---Dan (speaking for myself, except that the ciphertext-size numbers are speaking for themselves)

From:	pqc-forum@list.nist.gov on behalf of Vadim Lyubashevsky <vadim.lyubash@gmail.com>
Sent:	Thursday, October 7, 2021 7:57 AM
To:	pqc-forum; pqc-comments
Subject:	Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

Dear all,

Dan wrote:

> Details follow, beginning with a review of ciphertext sizes and
> continuing with replies to the "OFFICIAL COMMENT".
>
> Regarding 2x ciphertext compression, the numbers speak for themselves:
>
>   * 1568-byte ciphertext, including suboptimal encoding and CCA
>   protection: Kyber-1024. This uses modulus q=3329.
>
>   * 2996-byte ciphertext, with optimal encoding, no CCA protection: LPR
>   with the ring (\Z/3329)[x]/(x^1024+1). An LPR ciphertext has 2 ring
>   elements, and ceil(2*1024*log(3329)/log(256)) = 2996.

Why is Kyber with compression being compared to a *different* scheme without compression? Without *any* compression, Kyber-1024 would have
256*12*5/8 =1920 bytes in the ciphertext. With the compression directly from [page 17 of https://web.eecs.umich.edu/~cpeikert/pubs/svpcrypto.pdf] where the entire ciphertext is rounded to the same precision, we can drop 2 bits from each coefficient of the ciphertext and end up with
256*10*5/8=1600 bytes. This precedes any patents. So the only question now is whether dropping a different number of bits from different parts of the ciphertext is something covered by Ding's patent or not. I already weighed in on this many times, so I won't repeat my argument again -- but just so it's clear to everyone, the argument pertaining the ciphertext size of Kyber-1024 and Ding's patent is an argument about 32 bytes (so 2% and not a factor of 2).

Best,
Vadim

>
> 1568 is 52% of 2996. Here's another example:
>
>   * 1472-byte ciphertext, with optimal encoding, including CCA
>   protection: FireSABER. This uses modulus q=8192.
>
>   * 3328-byte ciphertext, with optimal encoding, no CCA protection:
>   LPR with the ring (\Z/8192)[x]/(x^1024+1).

From: pqc-forum@list.nist.gov on behalf of Christopher J Peikert <cpeikert@alum.mit.edu>
Sent: Thursday, October 7, 2021 11:16 AM
To: Vadim Lyubashevsky
Cc: pqc-forum; pqc-comments
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

On Thu, Oct 7, 2021 at 7:58 AM Vadim Lyubashevsky <vadim.lyubash@gmail.com> wrote:

> Dear all,
>
> Dan wrote:
>
> > Details follow, beginning with a review of ciphertext sizes and
> > continuing with replies to the "OFFICIAL COMMENT".
> >
> > Regarding 2x ciphertext compression, the numbers speak for themselves:
> >
> > * 1568-byte ciphertext, including suboptimal encoding and CCA
> > protection: Kyber-1024. This uses modulus q=3329.
> >
> > * 2996-byte ciphertext, with optimal encoding, no CCA protection: LPR
> > with the ring (\Z/3329)[x]/(x^1024+1). An LPR ciphertext has 2 ring
> > elements, and ceil(2*1024*log(3329)/log(256)) = 2996.
>
> Why is Kyber with compression being compared to a *different* scheme
> without compression? Without *any* compression, Kyber-1024 would have
> 256*12*5/8 =1920 bytes in the ciphertext.

Vadim is of course correct. The actual difference between compressed and uncompressed Kyber-1024 is about 1.22x. The numbers do indeed speak for themselves: this is not even close to 2x (it's less than a third of the way there).

That this is the first (and best?) attempted public justification ever offered for the FAQ's "2x" claim is extremely telling. But the claim is factually false, so it's going to have a hard time.

More broadly, **all** of the offered comparisons between Kyber/SABER and LPR are nonsensical, because of the significant difference in size between Module-LWE/LWR-style (like Kyber/SABER) and Ring-LWE-style (like LPR) ciphertexts. The comparisons misleadingly suggest that this difference is due to the cited patent instead of to its actual source, the use of Module-LWE/LWR.

(For the record, the patent says nothing about Module-LWE-style constructions, and Module-LWE was first defined in prior art from 2011, under the name "General LWE": https://eprint.iacr.org/2011/277 .)

The easy calculation: ignoring the (small, generic) CCA overhead,

1. An uncompressed "LPR-1024" ciphertext is **two elements** of a **1024-dimensional** ring mod q. This uses one full ring element to "carry" the encapsulated key.
2. An uncompressed Kyber-1024 ciphertext is **five elements** of a **256-dimensional** ring mod q. This is due to Kyber's use of Module-LWE over a 256-dimensional ring, with only one smaller ring element needed to carry the key.

Of course, 5*256 = 1280 is less than 2*1024=2048, by an 8/5 = 1.6x factor. This is much more than the 1.22x factor obtained from the drop-some-low-bits compression. (Similar comments apply to SABER as well.)

With the "2x" claim now conclusively refuted, I stand by my original comments.

Sincerely yours in cryptography,
Chris
--

From:	'Moody, Dustin (Fed)' via pqc-forum <pqc-forum@list.nist.gov>
Sent:	Thursday, October 7, 2021 11:52 AM
To:	Christopher J Peikert
Cc:	pqc-forum
Subject:	Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

Chris Peikert asked about what NIST can do about behavior that is "outside the bounds of fair play," and offered some suggestions. We appreciate that. We will do our best to set a standard of good behavior by example. The goal of creating secure post-quantum standards for the future should remain the primary focus of the pqc-forum. We have always encouraged interested parties to ask questions, make comments on the submissions, share their relevant work, etc. in a civil and constructive way.

NIST appreciates the dialogue fostered on this forum, and understands that disagreements are a natural part of the process. However, unethical behavior can undermine the integrity of the PQC process. Politely and professionally addressing misleading or inaccurate statements is an appropriate part of maintaining high academic and ethical standards.

NIST's four core values are perseverance, integrity, inclusivity, and excellence. Towards these values, NIST strongly discourages dishonesty, misrepresentations of science, personal attacks, and any form of hostility. The IEEE Code of Ethics https://www.ieee.org/about/corporate/governance/p7-8.html encapsulates the ethical standard we expect from ourselves and the community as we continue the PQC Standardization process.

NIST realizes there are limits to the time and energy that the community can realistically spend on "calling out" questionable behavior. There is also only so much that NIST can do. We respect people's right to share their work and opinions in a scientific way. When somebody is acting in a negative or malicious manner, it doesn't strengthen their argument, and we think most of the community will discount what they are saying (and we tend to do the same).

In that spirit, the NIST PQC team disagrees with some of the statements in the NTRUprime FAQ. For example, as Chris noted, we have not been discouraging public discussion on patent issues that may be relevant to the PQC standardization process. Also, we disagree that we have been inconsistent in handling security categories, or any suggestion that we are favoring one submission more than another. Ultimately, NIST will need to select the most promising algorithms for standardization, but we try to treat each submission in the same fashion. (These are just 2 examples - this doesn't mean we agree or disagree with everything else in the FAQ.)

Dustin Moody
NIST

From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> on behalf of Christopher J Peikert <cpeikert@alum.mit.edu>
Sent: Thursday, October 7, 2021 11:16 AM
To: Vadim Lyubashevsky <vadim.lyubash@gmail.com>
Cc: pqc-forum <pqc-forum@list.nist.gov>; pqc-comments <pqc-comments@nist.gov>
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

On Thu, Oct 7, 2021 at 7:58 AM Vadim Lyubashevsky <vadim.lyubash@gmail.com> wrote:

> Dear all,
>
> Dan wrote:
>
> > Details follow, beginning with a review of ciphertext sizes and
> > continuing with replies to the "OFFICIAL COMMENT".
> >
> > Regarding 2x ciphertext compression, the numbers speak for themselves:
> >
> >   * 1568-byte ciphertext, including suboptimal encoding and CCA
> >   protection: Kyber-1024. This uses modulus q=3329.
> >
> >   * 2996-byte ciphertext, with optimal encoding, no CCA protection: LPR
> >   with the ring (\Z/3329)[x]/(x^1024+1). An LPR ciphertext has 2 ring
> >   elements, and ceil(2*1024*log(3329)/log(256)) = 2996.
>
> Why is Kyber with compression being compared to a *different* scheme
> without compression? Without *any* compression, Kyber-1024 would have
> 256*12*5/8 =1920 bytes in the ciphertext.

Vadim is of course correct. The actual difference between compressed and uncompressed Kyber-1024 is about 1.22x. The numbers do indeed speak for themselves: this is not even close to 2x (it's less than a third of the way there).

That this is the first (and best?) attempted public justification ever offered for the FAQ's "2x" claim is extremely telling. But the claim is factually false, so it's going to have a hard time.

More broadly, **all** of the offered comparisons between Kyber/SABER and LPR are nonsensical, because of the significant difference in size between Module-LWE/LWR-style (like Kyber/SABER) and Ring-LWE-style (like LPR) ciphertexts. The comparisons misleadingly suggest that this difference is due to the cited patent instead of to its actual source, the use of Module-LWE/LWR.

(For the record, the patent says nothing about Module-LWE-style constructions, and Module-LWE was first defined in prior art from 2011, under the name "General LWE": https://eprint.iacr.org/2011/277 .)

The easy calculation: ignoring the (small, generic) CCA overhead,

From:	pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent:	Friday, October 8, 2021 8:18 AM
To:	pqc-forum
Subject:	Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments:	signature.asc

'Moody, Dustin (Fed)' via pqc-forum writes:
> For example, as Chris noted, we have not been discouraging public
> discussion on patent issues that may be relevant to the PQC
> standardization process.

Moody, Dustin (Fed) writes:
> Subject: IP statements
> From: "Moody, Dustin (Fed)" <dustin.moody@nist.gov>
> Date: Mon, 7 Dec 2020 15:24:16 +0000
> To: "D. J. Bernstein" <djb@cr.yp.to>
> Message-ID:
> <BLAPR09MB60814E534E6A24BB88035129E5CE0@BLAPR09MB6081.namprd09.prod.ou
> tlook.com>
>
> Dan,
>
> Regarding some of your comments on the forum regarding patents:
>
> We note that your submitted primitive NTRU-LPRime is a structured
> lattice scheme that uses rounding and truncation during the encryption
> process in a similar way to Kyber and Saber. We further note that the
> owners of your submission, NTRUprime, which incorporates NTRU-LPRime,
> are listed as the same as the submitters, and your latest signed IPR
> statement does not list any patents covering your submission. Does any
> of this need to be revised based on the statements you are making on the forum?
>
> We are working to clear up the IP situation, but it is a slow process.
> We hope everyone will focus on the technical issues, rather than on
> the patents right now.
>
> Dustin

From:	pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent:	Saturday, October 9, 2021 12:55 AM
To:	pqc-forum
Subject:	Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments:	signature.asc

I must admit to being surprised that, after I sent the message below, the whole business day at NIST went by without NIST issuing apologies for the evident misinformation and the underlying pressure tactic. I understand that a few days might be needed for issuing a post-mortem.

---Dan

D. J. Bernstein writes:
> Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
> From: "D. J. Bernstein" <djb@cr.yp.to>
> Date: Fri, 8 Oct 2021 14:17:37 +0200
> To: pqc-forum@list.nist.gov
> Message-ID: <20211008121737.6212.qmail@cr.yp.to>
>
> 'Moody, Dustin (Fed)' via pqc-forum writes:
> > For example, as Chris noted, we have not been discouraging public
> > discussion on patent issues that may be relevant to the PQC
> > standardization process.
>
> Moody, Dustin (Fed) writes:
> > Subject: IP statements
> > From: "Moody, Dustin (Fed)" <dustin.moody@nist.gov>
> > Date: Mon, 7 Dec 2020 15:24:16 +0000
> > To: "D. J. Bernstein" <djb@cr.yp.to>
> > Message-ID:
> > <BLAPR09MB60814E534E6A24BB88035129E5CE0@BLAPR09MB6081.namprd09.prod.
> > outlook.com>
> >
> > Dan,
> >
> > Regarding some of your comments on the forum regarding patents:
> >
> > We note that your submitted primitive NTRU-LPRime is a structured
> > lattice scheme that uses rounding and truncation during the
> > encryption process in a similar way to Kyber and Saber. We further
> > note that the owners of your submission, NTRUprime, which
> > incorporates NTRU-LPRime, are listed as the same as the submitters,
> > and your latest signed IPR statement does not list any patents
> > covering your submission. Does any of this need to be revised based on the statements you are making on the forum?
> >
> > We are working to clear up the IP situation, but it is a slow

From: pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent: Monday, October 11, 2021 3:20 AM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments: signature.asc

Email from NIST dated 7 Oct 2021 15:51:54 +0000 states, regarding the NISTPQC evaluation process, that NIST tends to discount input from people "acting in a negative or malicious manner". I have five clarification questions regarding this process, and I would appreciate having the answers numbered accordingly.

1. How does NIST evaluate which people are acting in a "negative" manner in the discounting context? For example, do the following personal attacks issued over the past year by Daniel Apon, "Kirk Fleming", Christopher J Peikert, Jacob Alperin-Sheriff, and Vadim Lyubashevsky qualify as "negative"? If no, why not?

* Daniel Apon (NIST employee), pqc-forum email dated 20 Jun 2021 16:04:49 -0700, personal attack without NIST speaking up to object: "Apparently everyone but you understands the state of the science, and is willing to accept new results as they happen. [new paragraph:] Stop propagandizing."

* "Kirk Fleming" (sockpuppet in violation of the transparency rules stated in https://www.nist.gov/system/files/documents/2017/05/09/VCAT-Report-on-NIST-Cryptographic-Standards-and-Guidelines-Process.pdf), pqc-forum email dated 24 Nov 2020 02:01:07 +0100, personal attack without NIST speaking up to object: "This is the irrelevant hype I mean. You are ignoring" etc.

* Christopher J Peikert, pqc-forum email dated 21 May 2021 09:30:17 -0400, personal attack without NIST speaking up to object: "... you have ignored the correction and persist in repeating the error. (My guess is that this is because it's central to sowing FUD-y confusion, by conflating the prior art and NIST candidates with 2012 Ding.)"

* Jacob Alperin-Sheriff (former NIST employee), pqc-forum email dated 22 Sep 2021 23:45:09 -0400, personal attack[1] without NIST speaking up to object: "Finally vindication for Daniel Bernstein has been achieved!" [footnote 1:] Given the context, many readers will see the message as a sarcastic attack (founded, I should note, upon misinformation regarding the history), whether or not it was intended this way. Even without the context, the gratuitous personalization is clear.

* Vadim Lyubashevsky, pqc-forum email dated 21 May 2021 12:59:46 +0200, personal attack without NIST speaking up to object: "It's that you're applying your usual anti-lattice crypto spin on things" etc.

2. How does NIST evaluate which people are acting in a "malicious"
manner in the discounting context? For example, given revelations of NSA working to undermine cryptographic standards, do NSA agents qualify as acting in a "malicious" manner? If no, why not?

3. When the discounting process is triggered, how does it work? What procedures are in place generally to protect against errors and abuse, and specifically to ensure that this process does not damage the NISTPQC evaluations mandated by 81 FR 92787 and by the call for submissions cited in 81 FR 92787?

4. When did this discounting process begin? Was it announced before October 2021? On what dates was it announced?

5. Where is this discounting approach to NISTPQC authorized in 81 FR
92787 or in the call for submissions cited in 81 FR 92787?

---Dan

From: Vadim Lyubashevsky <vadim.lyubash@gmail.com>
Sent: Monday, October 11, 2021 6:23 AM
To: pqc-forum; pqc-comments
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

On Mon, Oct 11, 2021 at 9:20 AM D. J. Bernstein <djb@cr.yp.to> wrote:

> 1. How does NIST evaluate which people are acting in a "negative"
> manner in the discounting context? For example, do the following
> personal attacks issued over the past year by Daniel Apon, "Kirk
> Fleming", Christopher J Peikert, Jacob Alperin-Sheriff, and Vadim
> Lyubashevsky qualify as "negative"? If no, why not?
>
> * Vadim Lyubashevsky, pqc-forum email dated 21 May 2021 12:59:46
> +0200, personal attack without NIST speaking up to object: "It's
> that you're applying your usual anti-lattice crypto spin on things"
> etc.

That does not qualify as negative. Given how many outright false things you've stated on this list about Kyber and lattice crypto (let's just take the concretely false and continued claim that uncompressed kyber is 2x longer than compressed kyber as the most recent example), I think that describing you as just having an "anti-lattice spin" was actually acting in a very positive manner.

Vadim

From:	D. J. Bernstein <djb@cr.yp.to>
Sent:	Monday, October 11, 2021 6:37 AM
To:	pqc-comments
Cc:	pqc-forum
Subject:	Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments:	signature.asc

Vadim Lyubashevsky writes:
> That does not qualify as negative.

I believe I was perfectly clear in asking questions about NIST's newly announced procedures. I would like authoritative answers from NIST regarding those procedures.

> Given how many outright false
> things you've stated on this list about Kyber and lattice crypto
> (let's just take the concretely false and continued claim that
> uncompressed kyber is 2x longer than compressed kyber as the most
> recent example),

Please either (1) quote where I said something about "uncompressed Kyber" or (2) withdraw this claim. Thanks in advance.

---Dan

From: Vadim Lyubashevsky <vadim.lyubash@gmail.com>
Sent: Monday, October 11, 2021 7:06 AM
To: pqc-forum; pqc-comments
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

On Mon, Oct 11, 2021 at 12:37 PM D. J. Bernstein <djb@cr.yp.to> wrote:
>
> Vadim Lyubashevsky writes:
> > That does not qualify as negative.
>
> I believe I was perfectly clear in asking questions about NIST's newly
> announced procedures. I would like authoritative answers from NIST
> regarding those procedures.

Sorry, I wanted to defend myself before NIST renders it's decision.
Promise not to interfere again.

> > Given how many outright false
> > things you've stated on this list about Kyber and lattice crypto
> > (let's just take the concretely false and continued claim that
> > uncompressed kyber is 2x longer than compressed kyber as the most
> > recent example),
>
> Please either (1) quote where I said something about "uncompressed
> Kyber" or (2) withdraw this claim. Thanks in advance.

Fair enough - I can admit when I am wrong. When you wrote:

"There are known patent threats against ... Kyber, SABER, and NTRU LPRime (ntrulpr). These proposals use ... and a 2x ciphertext-compression mechanism that appears to be covered by U.S. patent 9246675 expiring 2033,"

you did not explicitly state that by using a 2x ciphertext-compression mechanism, kyber actually compresses its ciphertext 2x. And when Chris asked you about it in this thread, you replied with:

"Regarding 2x ciphertext compression, the numbers speak for themselves:

* 1568-byte ciphertext, including suboptimal encoding and CCA
  protection: Kyber-1024. This uses modulus q=3329.

* 2996-byte ciphertext, with optimal encoding, no CCA protection: LPR
  with the ring (\Z/3329)[x]/(x^1024+1). An LPR ciphertext has 2 ring
  elements, and ceil(2*1024*log(3329)/log(256)) = 2996."

So there is a 2X ciphertext compression, but just not between
compressed and uncompressed Kyber. So I withdraw my claim about you
explicitly saying anything about uncompressed kyber and, based on the above examples, also relegate my statement about you "putting an anti-lattice crypto spin on things" from being positive to being merely accurate.

Vadim

>
> ---Dan
>
> --
> 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/20211011103709.179679.qmail%40cr.yp.to.

From: pqc-forum@list.nist.gov on behalf of Christopher J Peikert <cpeikert@alum.mit.edu>
Sent: Monday, October 11, 2021 8:11 AM
To: pqc-forum; pqc-comments
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

On Mon, Oct 11, 2021 at 3:20 AM D. J. Bernstein <djb@cr.yp.to> wrote:

> * Christopher J Peikert, pqc-forum email dated 21 May 2021 09:30:17 -0400, personal attack without NIST speaking up to object: "... you have ignored the correction and persist in repeating the error. (My guess is that this is because it's central to sowing FUD-y confusion, by conflating the prior art and NIST candidates with 2012 Ding.)"

This is not a "personal attack" [1], nor an "ad hominem attack" [2] (a term you recently used in another thread).

It is an evidence-based criticism of specific actions and statements (repeating factual errors after ignoring corrections, and conflating prior art with a patent, i.e., a form of FUD).

I would make the same criticism of anyone else who exhibited this behavior; you just happened to be the one to do it.

[1] "An abusive remark on or relating to somebody's person instead of providing evidence when examining another person's claims or comments." https://en.wiktionary.org/wiki/personal_attack
[2] https://en.wikipedia.org/wiki/Ad_hominem

The above, however, is a good reminder about the original and central topic of this thread [link].

It has been more than 10 months since the official NTRU Prime FAQ alleged that Kyber and SABER use "a 2x ciphertext-compression mechanism that appears to be covered by" a patent.

For months, this "2x" claim was pointed out as being factually false on multiple occasions, and no attempt to substantiate it even emerged until a few days ago.

This attempted substantiation was based on a nonsensical comparison between compressed Kyber/SABER and a different cryptosystem over a different ring, rather than between Kyber/SABER with and without ciphertext compression.

(For Kyber, this compression is the use of Compress/Decompress on the ciphertext components (u,v), whose main purpose "is to be able to discard some low-order bits in the ciphertext which do not have much effect on the correctness probability of decryption–thus reducing the size of ciphertext").

Very soon after, it was conclusively shown that the compression is only about 1.22x for the considered Kyber parameters---not even a third of the way to 2x.

At the time of this writing, the false "2x" claim remains on the NTRU Prime FAQ.

Sincerely yours in cryptography,
Chris

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Monday, October 11, 2021 8:53 AM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments: signature.asc

A 2010 LPR ciphertext has 2 ring elements, around 3 kilobytes for 2^256 Core-SVP when parameters are chosen to optimize ciphertext size.

A patented 2012 Ding ciphertext has just 1+epsilon, around 1.5 kilobytes for 2^256 Core-SVP. That's 2x compression compared to the relevant baseline, namely LPR.

2014 Peikert achieved the same 1+epsilon, and said that doing better than the LPR ciphertext size ("two R_q elements") was unobvious, the result of an "innovation". In court this is a slam-dunk against retroactive claims of obviousness for smaller ciphertexts than LPR.

Replacing rings with modules _in Ding's system_ makes negligible difference in ciphertext size and will be covered by the doctrine of equivalents. Avoiding "reconciliation" is ill-defined, so it won't even reach a doctrine-of-equivalents analysis. Trying to avoid Ding's patent by mixing Ding's technique with other techniques can only work if a Markman hearing ends up with a narrow interpretation of the claims, and I see nothing that can be used to justify such an interpretation.

The NTRU Prime FAQ correctly reports this:

> There are known patent threats against the "Product NTRU"/"Ring-LWE"/"LPR" lattice proposals: Kyber, SABER, and NTRU LPRime (ntrulpr). These proposals use a "noisy DH + reconciliation" structure that appears to be covered by U.S. patent 9094189 expiring 2032, and a 2x ciphertext-compression mechanism that appears to be covered by U.S. patent 9246675 expiring 2033. There are also international patents, sometimes with different wording.

The baseline for "2x" is, obviously, LPR. Kyber compresses its ciphertexts 2x compared to LPR. Recent references on pqc-forum to "uncompressed Kyber" are irrelevant to the history, irrelevant to a patent case, and irrelevant to the FAQ; the notion that the FAQ is comparing Kyber to "uncompressed Kyber" is obviously incorrect. The claim that there has been a "concretely false and continued claim that uncompressed kyber is 2x longer than compressed kyber" is incorrect and, unfortunately, not properly withdrawn.

---Dan

From:	pqc-forum@list.nist.gov on behalf of Christopher J Peikert <cpeikert@alum.mit.edu>
Sent:	Monday, October 11, 2021 9:34 AM
To:	pqc-forum; pqc-comments
Subject:	Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

On Mon, Oct 11, 2021 at 8:53 AM D. J. Bernstein <djb@cr.yp.to> wrote:

> Replacing rings with modules _in Ding's system_ makes negligible
> difference in ciphertext size and will be covered by the doctrine of
> equivalents.

This is irrelevant to the present discussion, because _it's not what Kyber/SABER do_.

They replace rings with modules _in the LPR system_, and do ciphertext compression using a drop-low-bits technique which does not appear in the patent, and which was described in at least four well known works of prior art, including for rings and modules (see, e.g., https://eprint.iacr.org/2011/277 and search for "modulus switching"). I have pointed this out multiple times, and have yet to see your arguments acknowledge it.

> Trying to avoid Ding's patent
> by mixing Ding's technique with other techniques

Also irrelevant, because _it's not what Kyber/SABER do_. They don't use Ding's technique at all, and it's wrong to imply that they do.

> The NTRU Prime FAQ correctly reports this:
>
> There are known patent threats against the "Product NTRU"/"Ring-LWE"/"LPR" lattice proposals: Kyber, SABER, and NTRU LPRime (ntrulpr). These proposals use a "noisy DH + reconciliation" structure that appears to be covered by U.S. patent 9094189 expiring 2032, and a 2x ciphertext-compression mechanism that appears to be covered by U.S. patent 9246675 expiring 2033. There are also international patents, sometimes with different wording.
>
> The baseline for "2x" is, obviously, LPR.

It's not obvious; it's not even plausible. For Kyber, the reader will see "a 2x ciphertext-compression mechanism" as referring to Kyber's Compress/Decompress algorithms, which are correctly described in the spec as the mechanism for compressing ciphertexts.

(Section "Compression and Decompression": "The main reason for defining the Compress_q and Decompress_q functions is ... reducing the size of ciphertexts.")

Sincerely yours in cryptography,
Chris

--

From:	Vadim Lyubashevsky <vadim.lyubash@gmail.com>
Sent:	Monday, October 11, 2021 9:35 AM
To:	pqc-forum; pqc-comments
Subject:	Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

> Replacing rings with modules _in Ding's system_ makes negligible
> difference in ciphertext size and will be covered by the doctrine of
> equivalents. Avoiding "reconciliation" is ill-defined, so it won't
> even reach a doctrine-of-equivalents analysis. Trying to avoid Ding's
> patent by mixing Ding's technique with other techniques

No one is mixing Ding's technique with anything. The encapsulation (including compression) in Kyber is the Encaps algorithm from page 17 of
https://web.eecs.umich.edu/~cpeikert/pubs/svpcrypto.pdf,
with the only difference that q' does not have to be the same for b_1 and b_2. That's it. I don't see how anything could be closer to what Kyber does without actually being Kyber; and if the doctrine of equivalents applies to Kyber from Ding's patent (who, as I mention again, did not compress the a public key encryption scheme in the same patent), then surely one should be able to apply it from the linked paper.

Vadim

From: pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent: Monday, October 11, 2021 10:26 AM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments: signature.asc

Christopher J Peikert writes:
> using a drop-low-bits technique which does not appear in the patent

See my pqc-forum message dated 1 Jan 2021 13:19:26 +0100 for a detailed analysis of the specific claims of a dividing line here.

> The NTRU Prime FAQ correctly reports this:
> There are known patent threats against the "Product
> NTRU"/"Ring-LWE"/"LPR" lattice proposals: Kyber, SABER, and NTRU
> LPRime (ntrulpr). These proposals use a "noisy DH + reconciliation"
> structure that appears to be covered by U.S. patent 9094189 expiring
> 2032, and a 2x ciphertext-compression mechanism that appears to be
> covered by U.S. patent 9246675 expiring 2033. There are also
> international patents, sometimes with different wording.
> The baseline for "2x" is, obviously, LPR.
> It's not obvious; it's not even plausible.

The text is clear: there's noisy DH + reconciliation (i.e., LPR), and then there's ciphertexts shrinking 2x.

---Dan

From: pqc-forum@list.nist.gov on behalf of Christopher J Peikert <cpeikert@alum.mit.edu>
Sent: Monday, October 11, 2021 12:29 PM
To: pqc-forum; pqc-comments
Subject: Re: [pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime

On Mon, Oct 11, 2021 at 10:26 AM D. J. Bernstein <djb@cr.yp.to> wrote:

Christopher J Peikert writes:
> using a drop-low-bits technique which does not appear in the patent

See my pqc-forum message dated 1 Jan 2021 13:19:26 +0100 for a detailed
analysis of the specific claims of a dividing line here.

Then see my response [link] showing a clear dividing line between the drop-some-low-bits prior art (which Kyber/SABER use) and the technique from the patent (which they don't).

>   The NTRU Prime FAQ correctly reports this:
>   There are known patent threats against the "Product
>   NTRU"/"Ring-LWE"/"LPR" lattice proposals: Kyber, SABER, and NTRU
>   LPRime (ntrulpr). These proposals use a "noisy DH + reconciliation"
>   structure that appears to be covered by U.S. patent 9094189 expiring
>   2032, and a 2x ciphertext-compression mechanism that appears to be
>   covered by U.S. patent 9246675 expiring 2033. There are also
>   international patents, sometimes with different wording.
>   The baseline for "2x" is, obviously, LPR.
> It's not obvious; it's not even plausible.

The text is clear: there's noisy DH + reconciliation (i.e., LPR), and
then there's ciphertexts shrinking 2x.

The text is indeed clear: the "2x" adjective is applied specifically to Kyber/SABER's "ciphertext-compression mechanism" of dropping some low bits.

Anyway, the facts have been laid out, and readers can judge for themselves whether this paragraph is an accurate characterization.

Sincerely yours in cryptography,
Chris
--

Kerman, Sara J. (Fed)

From:	pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent:	Sunday, October 31, 2021 5:43 PM
To:	pqc-comments
Cc:	pqc-forum
Subject:	[pqc-forum] ROUND 3 OFFICIAL COMMENT: NTRU Prime
Attachments:	latticerisks-20211031.pdf; signature.asc

[This is an official comment regarding all of the lattice KEMs, but to avoid repetition is being filed just once.]

The attached document "Risks of lattice KEMs" (also available from https://ntruprime.cr.yp.to/warnings.html)

* surveys recent attack advances;

* classifies ongoing risks, fully defining the risk table shown in https://ntruprime.cr.yp.to/warnings.html;

* reviews incorrect claims that proofs control these risks;

* analyzes performance, since performance issues can exacerbate security risks; and

* compares the KEMs according to the official NISTPQC evaluation criteria.

The document is authored by the NTRU Prime Risk-Management Team and is hereby added to the NTRU Prime submission. The document has the following abstract:

Lattice-based KEMs under consideration within the NIST Post-Quantum Cryptography Standardization Project (NISTPQC) are much more risky than commonly acknowledged. In applications where performance constraints force the use of a lattice-based KEM, the least risky option available is NTRU Prime, specifically Streamlined NTRU Prime (sntrup) at the largest size that fits those performance constraints.

---Dan

Risks of lattice KEMs

NTRU Prime Risk-Management Team

Abstract. Lattice-based KEMs under consideration within the NIST Post-Quantum Cryptography Standardization Project (NISTPQC) are much more risky than commonly acknowledged. In applications where performance constraints force the use of a lattice-based KEM, the least risky option available is NTRU Prime, specifically Streamlined NTRU Prime (`sntrup`) at the largest size that fits those performance constraints.

1 Introduction: Cryptanalytic overload

Years from now, when we're all looking back at today's NIST Post-Quantum Cryptography Standardization Project (NISTPQC) and asking "How did the U.S. government manage to standardize a post-quantum KEM that was then shown to be broken?", surely a large part of the answer is going to be that, even though we have *some* people publicly looking for attacks, the post-quantum attack surface is so much bigger than this that we ran out of time.

The broken standard is still a hypothetical scenario at this point. There is, however, already ample evidence of the cryptanalytic overload. Consider the following two examples:

- Round2, a "provably secure" lattice system submitted to round 1 of NIST-PQC in 2017, was broken by a very fast attack [**39**] from 2020 Bellare–Davis–Günther. The attack exploits the pattern of hash inputs inside the FO transform used inside Round2 to try to achieve CCA security; a mistake in these details can easily destroy all security.
- The official Frodo software from 2017—software that the submission [**15**, Section 3.1] claimed was "protected against timing and cache attacks"—was broken by a feasible timing attack [**169**] from 2020 Guo–Johansson–Nilsson. The attack exploits the fact that the software used `memcmp`, a subroutine well known to take variable time.

Why were these vulnerabilities not announced in 2017? 2018? 2019? These are not subtle mistakes. Presumably a large-scale attacker, having already hired and trained thousands of people to look for attacks, would have found these mistakes immediately after the submissions were posted in 2017—but there are far fewer people *publicly* working on attacks, and the attack surface is massive, so easy attacks took years to discover.

Some people will tell you something like this: "These two attacks are nothing to worry about. We're working on formal verification, which any day now is

Permanent ID of this document: `ca26cf30f192b2705c98b025b9a51d68d735e694`.
Date: 2021.10.31.

submission	NTRU		NTRU Prime		SABER	Kyber	Frodo
KEM family	`ntruhrss`	`ntruhps`	`sntrup`	`ntrulpr`	`saber`	`kyber`	`frodo`
KEM	701	40961229	1277	1277	`fire`	1024	640
Basics							
pk+ct bytes	2276	3684	3914	3822	2784	3136	19336
ct bytes	1138	1842	1847	1975	1472	1568	9720
errors	Noisy	Noisy	Rounded	Rounded	Rounded	Noisy	Noisy
Q or P	Quotient	Quotient	Quotient	Product	Product	Product	Product
modulus	8192	4096	7879	7879	8192	3329	32768
dimension	700	1228	1277	1277	1024	1024	640
first 2-norm	21.6	22.6	22.2	20.7	39.2	32.0	52.4
second 2-norm	21.6	28.6	29.2	29.2	75.0	32.0	52.4
$\log_2$(Core-SVP)	136	301	270	271	260	256	150
ct/$\log_2$ ratio	8.4	6.1	6.8	7.3	5.7	6.1	64.8
Known attack avenues not ruled out by theorems							
lattices	risk	risk	risk	risk	risk	risk	risk
derandomization				risk	risk	risk	risk
decryption failures					165	174	138
structured lattices	risk	risk	risk	risk	risk	risk	
cyclotomics	risk	risk			risk	risk	
reducibility	risk	risk			risk	risk	
quotients	risk	risk	risk				
extra samples				risk	risk	risk	risk
non-QROM FO	risk	risk	risk	risk	risk	risk	risk
non-QROM 2				risk	risk	risk	risk
Known patent threats							
patent 9094189				risk	risk	risk	
patent 9246675				risk	risk	risk	
Systemic risks							
PKE instability	2019.04	2019.04	2016.05	2017.12	2019.04	2020.10	2019.04
instability	2019.04	2019.04	2019.04	2019.04	2019.04	2020.10	2019.04

Table 1.1. Risks of proposed lattice-based KEMs. Risks of implementations deviating from specifications and risks of side-channel attacks are not included. See Section 3 for definition of each row. Each column takes largest specified KEM size that fits public key plus ciphertext into 4KB, except that the `frodo` column allows 20KB.

going to magically eliminate all of the CCA problems and all of the software problems. This means that the security of the software will follow from the core mathematical security of the lattice PKEs inside these KEMs. What matters is that we understand this core mathematical security: we know how hard it is to break SVP, how hard it is to break Approximate-SVP, and how hard it is to break these lattice PKEs."

Unfortunately, no, we *don't* know how hard it is to break these core lattice problems. The following list of attacks published *after* the NISTPQC submission deadline shows how unstable the attack picture is for SVP, and for Approximate-SVP, and for these lattice PKEs:

- 2018 Laarhoven–Mariano [218] saved "between a factor 20 to 40 in the time complexity for SVP".
- 2018 Bai–Stehlé–Wen [31] introduced a new variant of BKZ producing "bases of better quality" for the "same cost" of SVP.
- 2018 Aono–Nguyen–Shen [18] adapted "recent quantum tree algorithms" to enumeration.[1]
- 2018 D'Anvers–Vercauteren–Verbauwhede showed [126] that "an attacker can significantly reduce the security of (Ring/Module)-LWE/LWR based schemes that have a relatively high failure rate" and [127] that for LAC-128 "the failure rate is 2^{48} times bigger than estimated".
- 2018 Hamburg [174] pointed out that the first published "provably secure" Round5 design had disastrously high decryption-failure rate, 2^{-55}.
- 2019 Pellet-Mary–Hanrot–Stehlé [311] broke through the previously claimed $\exp(n^{1/2+o(1)})$ approximation-factor "barrier" for number-theoretic attacks against Ideal-SVP.
- 2019 Guo–Johansson–Yang [170] presented faster attacks against some systems that use error correction to (try to) reduce decryption failures. This paper violated the security claims of LAC.
- 2020 Dachman-Soled–Ducas–Gong–Rossi [128] presented slightly faster attacks against the constant-sum secrets used in three lattice submissions: LAC, NTRU, and Round5.
- 2020 Doulgerakis–Laarhoven–de Weger [136] presented "faster [sieving] methods" for SVP.
- 2020 Albrecht–Bai–Fouque–Kirchner–Stehlé–Wen [9] reduced the exponent of enumeration from $\approx 0.187\beta \log_2 \beta$ to $\approx 0.125\beta \log_2 \beta$. This improvement, in combination with [18], reduces the post-quantum security levels of a wide range of proposed lattice-based systems.[2]
- 2020 Albrecht–Bai–Li–Rowell [10] introduced a "practical and faster" enumeration algorithm "for reaching the same RHF in practical and cryptographic parameter ranges". This further reduces the post-quantum security levels of proposed lattice-based systems.
- 2020 Bernard–Roux-Langlois [40] improved the algorithm from [311], and showed experimentally that in small dimensions the improved algorithm reaches much better approximation factors.

[1] There is a misconception that enumeration is superseded by sieving for cryptographic sizes. In fact, because of [18], [9], and [10], the fastest quantum attacks known today use enumeration for all dimensions up to a cutoff *larger* than many of the KEMs proposed for deployment. The cutoff becomes even higher in metrics that account for communication cost, since sieving is much more memory-intensive than enumeration. See Appendix B.4.

[2] NIST's low-precision "categories" (see Appendix B) hide this security loss: they say that there is no change in "category" since known quantum speedups against lattice systems are still not as dramatic as the Grover speedup against AES. However, what matters for understanding the instability of the lattice attack picture is that lattice systems now have quantitatively lower post-quantum security levels than they did before, as a direct result of a dramatic improvement in enumeration speed.

- 2021 Bi–Lu–Luo–Wang–Zhang [76] introduced a hybrid dual attack that improves "the state-of-the-art cryptanalysis results by 2–14 bits, under the BKZ-core-SVP model".
- 2021 D'Anvers–Batsleer [125] improved the "state-of-the-art multitarget failure boosting attacks", showing that "the quantum security of Saber can theoretically be reduced from 172 bits to 145 bits in specific circumstances".
- 2021 May [257] improved combinatorial attacks from exponent $0.5 + o(1)$ to exponent $0.25 + o(1)$ in the case of ternary keys; 2021 van Hoof–Kirshanova–May [180] improved the exponents of quantum combinatorial attacks; 2021 Kirshanova–May [208] improved the $o(1)$. These combinatorial attacks are now the state-of-the-art attacks against some pre-NISTPQC proposals of lattice-based PKEs, including a proposal standardized by IEEE.
- 2021 Chailloux–Loyer [99] improved quantum sieving exponents by 3%, from $0.2653\ldots + o(1)$ to $0.257\ldots + o(1)$. Previously $0.2653\ldots + o(1)$ was believed to be optimal.[3]
- 2021 Heiser [176] introduced another quantum-sieving speedup that "affects the security of lattice-based encryption schemes, including NIST PQC Round 3 finalists", and conjectured that this speedup can be combined with [99].

1.2. Overconfidence. The impressive history of advances in lattice attacks is accompanied by an equally impressive history of displays of confidence that the advances had already reached their limits. For example, let's look at more of the history of number-theoretic attacks, specifically unit attacks against Ideal-SVP and their generalization to S-unit attacks:

- A breakthrough unit attack, combining a fast cyclotomic reduction algorithm by 2014 Campbell–Groves–Shepherd [96] with a fast quantum algorithm by Biasse–Song [78], extracts secret keys from the cyclotomic case (assuming "$h^+ = 1$") of well-known cryptosystems introduced by Gentry [161], Smart–Vercauteren [339], Gentry–Halevi [162], and Garg–Gentry–Halevi [157].
- [123, Section 1] stated that "the above-described algorithms … apply only to *principal* ideals" and described this as a "barrier". However, 2017 Cramer–Ducas–Wesolowski [124] showed, for cyclotomics, how to reach approximation factor $\exp(n^{1/2+o(1)})$ in polynomial time for *arbitrary* ideals.
- This approximation factor $\exp(n^{1/2+o(1)})$ was described in (for example) [307, PDF page 84], [306, minutes 61–62], [304, PDF page 75], and [309, minute 45] as a "barrier", a "natural barrier", and an "inherent barrier" for this line of work. However, as mentioned above, 2019 Pellet-Mary–Hanrot–Stehlé [311] broke through this "barrier". The algorithm of [311] reaches, for example, approximation factor just $\exp(n^{1/4+o(1)})$ in time $\exp(n^{1/2+o(1)})$, although it uses $\exp(n^{1+o(1)})$ precomputation.

[3] Consider [207] saying that "one can view our lower bounds on sieving with nearest neighbor searching as a further motivation for most concrete parameter selection methods currently used in practice, which assume that the leading time complexity exponents 0.292 and 0.265 are the best an attacker can do".

- A model of concrete sizes of these attacks was presented in [**138**] and claimed to be "somewhat reassuring for NIST candidates"; [**138**] dismissed [**311**] as using "an exponential amount of precomputation". However, a recent talk given by Bernstein [**60**] introduced a variety of advances in S-unit attacks, including much faster precomputation; presented publicly verifiable (π-digit) experiments in various sizes that had been presented in [**138**], showing these attacks finding much shorter vectors than indicated in [**138**]; and conjectured subexponential scalability.
- The conjecture of subexponential scalability was disputed in [**137**], which explained that applying a "standard heuristic" to S-unit lattices produced the conclusion that the probability of success of the attack in [**60**] "would be *ridiculously* small", certainly no better than [**311**]. It's correct that this is a standard heuristic in the literature on lattice-based cryptography, but applying this heuristic to S-unit lattices is an error, and the conclusion of [**137**] is wrong. See [**72**].

Given how many "barriers" (and "lower bounds" and so on) have been broken, one is forced to conclude that the risk-assessment mechanisms used in lattice-based cryptography are deeply flawed.

1.3. Consequences for NISTPQC. NIST's official evaluation criteria—see Appendix A for (1) the full criteria and (2) comparison of KEMs under those criteria—state that "The security provided by a cryptographic scheme is the most important factor in the evaluation". The criteria recognize that a complicated, unstable, inadequately understood area of cryptography is a security risk:

> As public-key cryptography tends to contain subtle mathematical structure, it is very important that the mathematical structure be well understood in order to have confidence in the security of a cryptosystem. To assess this, NIST will consider a variety of factors. All other things being equal, simple schemes tend to be better understood than complex ones. Likewise, schemes whose design principles can be related to an established body of relevant research tend to be better understood than schemes that are completely new, or schemes that were designed by repeatedly patching older schemes that were shown vulnerable to cryptanalysis.

Modern lattice-based cryptography is the result of many years of patching older schemes shown vulnerable to cryptanalysis. The most obvious patches are continued increases in lattice dimensions, reacting to demonstrations that older dimensions were too small to resist attack. Consider, e.g., the original NTRU proposal by 1996 Hoffstein–Pipher–Silverman estimating 2^{80} security for 104-byte public keys using lattice dimension 83 (see [**177**, Table 2, last column]). As another example, consider 2011 Lindner–Peikert [**232**, Section 1.1] stating that its system using key sizes of "400 kilobits" (with a public randomness source—see Section 3.16) appeared to be "at least as secure as AES-128", and that using rings "we can immediately shrink the above key sizes by a factor of at least 200".

Today it is well known that a 2-kilobit (256-byte) key in the ring version of the Lindner–Peikert system is not secure.

The NISTPQC submission deadline was in 2017, just six years after [**232**]—not very long from a cryptanalyst's perspective, especially for a complex topic. Various submissions claimed that lattice attacks were "well studied", and tried to use the switch from original NTRU to Ring-LWE/Module-LWE to separate themselves from the break-and-patch history of the area. In fact, this switch had very little effect on the attacks in the literature; for a unified survey see [**65**, Section 6]. What had much more of an effect was the switch to much larger lattice dimensions—but lattice attacks then continued to advance, reducing security levels of all lattice submissions and breaking some lattice submissions outright.

The Kyber submission portrayed its security analysis as "conservative" [**24**, pages 20–21] and gave a five-step argument [**24**, Section 4.4] concluding that it "seems clear" that breaking `kyber512`, Core-SVP 2^{112}, is harder than brute-force AES-128 key search. In 2020, [**54**] disproved part of the argument and showed that recent attack advances had undermined the rest of the argument. The round-3 Kyber submission then presented new `kyber512` security claims relying on

- **replacing `kyber512` with a patched cryptosystem** [**25**, page 2] and
- presenting a new analysis [**25**, pages 25–27] estimating that attacks against the *new* round-3 `kyber512` would use $2^{151.5}$ "gates", more than the estimated 2^{143} "gates" for AES-128 key search.

The Kyber presentation at the Third PQC Standardization Conference [**26**, video, 1:29–1:30] described the new $2^{151.5}$ estimate as "tentative", stating that we "need more research into this" and that there are "foreseeable improvements". A much broader range, from $2^{135.5}$ "gates" to $2^{165.5}$ "gates", appeared in [**26**] without the "tentative" label—but $2^{135.5}$ is not good enough[4] if NIST requires at least 2^{143} "gates". "Conservative" in 2017; bleeding-edge patches in 2020.

When one asks for evidence that the security of the proposed lattice KEMs is "well studied", the typical response is to say that there is a long history of papers studying lattice security. For example, 2006 Silverman [**337**] claimed that SVP had already been "intensively studied for more than 100 years". Unfortunately, the papers on attack algorithms keep showing *losses* of lattice security. Lattice security is "well studied" in the same sense that RC4 is "well studied": yes, all of the papers [**163**], [**210**], [**264**], [**151**], [**164**], [**150**], [**250**], [**263**], [**300**], [**301**], [**248**], [**249**], [**297**], [**295**], [**348**], [**6**], [**35**], [**79**], [**165**], [**202**], [**209**], [**243**], [**244**], [**255**], [**288**], [**298**], [**36**], [**254**], [**266**], [**296**], [**107**], [**108**], [**109**], [**267**], [**334**], [**358**], [**110**], [**111**], [**130**], [**172**], [**245**], [**274**], [**335**], [**344**], [**14**], [**112**], [**183**], [**234**], [**246**], [**242**], [**286**], [**173**], [**171**], [**184**], [**185**], [**276**], [**293**], [**326**], [**158**], [**186**], [**287**], [**327**], [**329**], [**347**], [**187**], [**188**], [**193**], [**328**], [**330**], [**353**], [**92**],

[4] NIST made the surprising and unexplained claim in December 2020 [**269**] that if the round-3 Kyber analysis is correct "then Kyber clearly meets the security categories defined in the CFP". Clarification questions [**57**] were ignored. NIST appears to be tweaking the boundaries of its "categories" to favor Kyber; see generally [**53**].

[132], [189], [192], [194], [294], [299], [100], [133], [102], [101], [103], [190] are studying the security of RC4, but this is fully explained by the fact that RC4 has many complicated weaknesses. Lattices also have many complicated weaknesses, and the list of recent attacks shows that the lattice attack picture has not stabilized.

1.4. Arguments to consider lattices despite the risks. Why are lattice-based cryptosystems being considered for standardization?

Answer #1 is pervasive advertising of the "provable security" of these systems. The reality, however, is that what has been proven by "provable security" is *not* security; it is something much weaker, controlling only certain corner risks and fundamentally not addressing the central risks in lattice-based cryptography. "Provable security" didn't stop any of the post-2017 advances in attacks, didn't save the broken systems, and won't stop further advances. See Section 5.

Answer #2 points to the ability of lattice-based cryptography to support fully homomorphic encryption, multilinear maps, etc. However:

- The concrete lattice-based KEMs proposed for standardization don't support any of these extra features.
- The recent attack literature shows that the security of systems with these features is even more poorly understood than the security of KEMs designed purely for IND-CCA2 security.

NIST says that being able to modify a scheme for extra functionality is good; however, users of broken lattice KEMs will not appreciate hearing that ample evidence of risks was submerged under advertisements of extra functionality of different lattice systems with even higher risks.

A better answer is that some applications seem[5] to be saying things like this: "I can only give you one kilobyte in my protocol for a ciphertext and one kilobyte for a public key. That's it." Frodo doesn't fit into 1KB. Classic McEliece ciphertexts fit very easily into 1KB, but the public keys are much larger. SIKE ciphertexts *and* public keys fit easily into 1KB, but Google refused to deploy SIKE because of SIKE's consumption of CPU time.[6] For applications with such performance constraints, one has to use a small lattice-based KEM. This creates a challenging question of how to manage the risks.

2 NTRU Prime: reducing attack surface at low cost

Let's assume that the application's performance requirements force the use of a small lattice system. Subject to this requirement, NTRU Prime is the only

[5] For example, Google [225] reported TLS interoperability problems with 4KB keys, although it's far from clear that this is a long-term issue. Experiments with 3300-byte keys were successful.

[6] This was reported in [217, page 26], regarding a joint Cloudflare-Google experiment with post-quantum cryptography: "Note: SIKE not cleared for Google servers due to DoS risk."

Design decisions didn't prioritize minimizing the attack surface

$\Downarrow$

Careful security review was much more complicated than necessary

$\Downarrow$

Community didn't have enough time to carefully review everything

$\Downarrow$

Nobody studied the relevant weakness in the attack surface

$\Downarrow$

Deployed post-quantum system turns out to be breakable

Fig. 2.1. Predicting post-quantum disasters. See Section 1 for a review of evidence that lattice-based cryptography is on this path, including examples of broken "provably secure" lattice submissions.

submission systematically designed to **eliminate unnecessary complications in security review**: eliminate decryption failures, eliminate cyclotomics, etc.

Does this help? At this point there is ample evidence that the answer is yes. Subsequent advances in lattice attacks fall into two classes: attacks that work against *all* small lattice systems, including NTRU Prime; and attacks that work against only *some* lattice systems, *not* including NTRU Prime—because NTRU Prime had already eliminated the tools used in those attacks. The path to disaster in Figure 2.1 is clear, and NTRU Prime has already demonstrated success at stopping this path at its first step.

2.2. Case study: decryption failures. There have been many advances in decryption-failure attacks, including recent advances listed in Section 1 that have broken the security claims for various NISTPQC submissions. For the schemes that haven't (yet?) been broken by decryption failures, there are more and more pages of increasingly complicated analysis. Consider, e.g., [**125**] saying "We first improve the state-of-the-art multitarget decryption failure attack using a levelled approach", pointing out "three inaccuracies in the directional failure boosting calculation for the simplified scheme of [11]", having to do more work because "this traditional approach of calculating the directional failure boosting cost is not directly applicable to practical schemes such as Kyber and Saber due to compression of the ciphertexts", etc.

Note that compression making an attack *analysis* difficult does not imply that it makes the *attack* difficult. One is reminded of a famous quote from Hoare:

> There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies.

Hoare's first way prioritizes minimizing the complexity of a thorough review.

The introduction of NTRU Prime in 2014 [**44**] had already noted that the problem of "figuring out whether an attacker can trigger decryption failures" was a "mess"—so it simply eliminated decryption failures. The proof that there are no failures takes just a few lines. A thorough review of the impact of decryption failures is vastly easier for NTRU Prime than for most lattice-based KEMs.

The introduction of `ntruhrss` in 2017 [**181**], like NTRU Prime, eliminated decryption failures; `ntruhps` switched to eliminating decryption failures when it merged with `ntruhrss` to form the round-2 NTRU submission. Kyber, SABER, and Frodo still have decryption failures, and this could damage their security; see Section 3.9.

2.3. Proactive vs. reactive. More broadly, the NTRU Prime approach is *proactively* looking at the attack surface, identifying attack tools that can be eliminated at the design stage, and eliminating them.

The alternative is reacting to systems getting broken: "That was broken, stop doing that". See Figure 2.4. History shows that this reactive approach is often triggered many years after the system was designed, because that's how long the public attack development took; see the next paragraph for an illustration of cryptanalytic time scales. It is important to realize that the system *was not secure* in the meantime: the public simply didn't know that it was insecure.

Consider the problem of key recovery for the DH cryptosystem using the group $\mathbf{F}_{2^n}^*$. Discrete-logarithm techniques that were already standard in the 1960s [**355**] take time $\exp(n^{1/2+o(1)})$; these are based on techniques introduced in the 1920s by Kraitchik [**213**], relying on how frequently integers are "smooth". A sudden breakthrough by 1984 Coppersmith [**120**] reduced 1/2 to 1/3. The cryptosystem can survive the attack of [**120**] by moving to somewhat larger n, but decades later this discrete-logarithm problem was publicly smashed by a new wave of papers, culminating in a devastating quasi-polynomial-time attack by 2014 Barbulescu–Gaudry–Joux–Thomé [**33**]. See the recent survey [**167**] for further history and a review of the attack tools, notably subfields and automorphisms.

Was the system secure before it was *publicly shown* to be insecure? No. Is it reasonable to extrapolate from the lengthy public development that *attackers didn't know the attack* until 2014? No. Consider the fact that the Institute for Defense Analyses, an NSA consulting company, many years ago hired Buhler, one of the original developers of the number-field sieve [**94**] for integer factorization; Gordon, the first developer of a discrete-logarithm version [**166**] of the number-field sieve; Miller, who as part of introducing ECC [**261**] was one of the first authors to probe the limits of discrete-logarithm algorithms; and Coppersmith. Much less data is available regarding the cryptanalytic capabilities of, e.g., the Chinese government. Surely large-scale attackers know many more attacks than this public does. A reactive approach is inherently incapable of protecting against these attacks, whereas a proactive approach has a chance.

2.5. Case study: cyclotomics. As a historical matter, most proposals of small lattice-based KEMs have used cyclotomic fields. These fields provide tools to the attacker that most number fields do not provide, such as subfields and automorphisms. 2013 Bernstein [**43**] noted the potential value of these subfields and automorphisms for finding, among other things, "short generators of ideals"

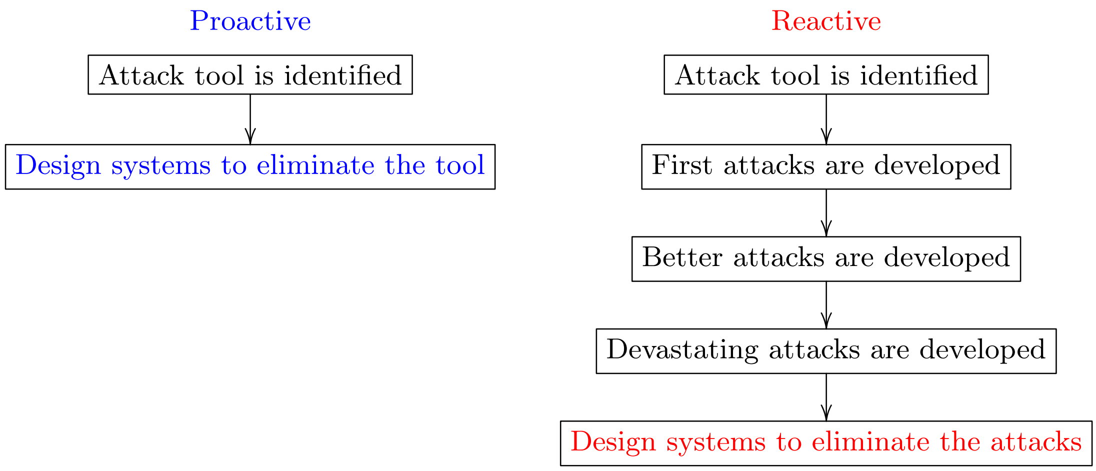

Fig. 2.4. Proactively handling advances in attacks vs. reactively handling advances in attacks.

and attacking "NTRU, Ring-LWE, FHE", and concluded that NTRU "should switch to random prime-degree extensions with big Galois groups".

Prime-degree fields are guaranteed to have no proper subfields other than $\mathbf{Q}$. Fields with "big Galois groups" are far from having automorphisms. Notice the proactive approach, taking tools away from the attacker *without* waiting for attacks to be developed. As an analogy, very similar recommendations to avoid subfields and automorphisms in discrete-log cryptography were published in [29] and [42] several years *before* quasi-polynomial-time discrete-log attacks were published using these attack tools.

A complete NTRU Prime cryptosystem was introduced in 2014 [44], as noted above. This cryptosystem uses specifically the field $\mathbf{Q}[x]/(x^p - x - 1)$. This might seem similar at first glance to the cyclotomic field $\mathbf{Q}[x]/((x^p - 1)/(x - 1))$ used in NTRU, but two centuries of development of algebraic number theory—see Section 2.6—have established vast mathematical gaps between these fields. For example:

- The Galois group of $\mathbf{Q}[x]/(x^p - x - 1)$ has size $p! \approx (p/e)^p$, the *maximum* possible size for a field of degree p.
- The Galois group of $\mathbf{Q}[x]/((x^p - 1)/(x - 1))$ has size $p - 1$, the *minimum* possible size for a field of degree $p - 1$.

Along with this cryptosystem, [44] introduced a "subfield-logarithm attack", improving on previous attacks in some cases. Later in 2014, Campbell–Groves–Shepherd [96] introduced a much faster attack in the case of cyclotomic fields. Followups broke various "barriers" claimed for this line of work; see Section 1.2. These breaks have taken more and more advantage of subfields, automorphisms, and further cyclotomic structure. See also the papers [37], [229], and [77] further developing subfield-logarithm attacks for various fields.

NIST has repeatedly[7] misstated the history here, in particular by describing NTRU Prime as a reaction to the attack of [96]. This description doesn't just get the order of events wrong; it actively hides (1) the cryptographer's ability to *proactively* protect against risks and (2) NTRU Prime's demonstrated success in doing this.

The situation since 2014 has been that some lattice-based cryptosystems are known to be broken for cyclotomic fields and *not* known to be broken for the NTRU Prime field, while there is nothing the other way around. (There are, of course, also some lattice-based cryptosystems known to be broken independently of the field choice, and some lattice-based cryptosystems not known to be broken either way.) The broken cyclotomic systems are clear evidence of a cyclotomic risk, and the tools used in the known attacks provide a clear explanation of where this risk is coming from. The risk-management conclusion is also clear: Stay away from cyclotomics.

2.6. Algorithms in algebraic number theory. Further notes are required on the mathematical context for NTRU Prime.

Algebraic number theory, the study of number fields, has been one of the primary research areas in number theory over the past two centuries. "A course in computational algebraic number theory", a well-known 1993 textbook [118] from Cohen (see also [119]), identifies five "main computational tasks of algebraic number theory", where three of them are computing the "unit group", computing the "class group", and computing generators of principal ideals. The book spends many pages describing sophisticated algorithms to compute unit groups, class groups, and generators; these topics are related to each other, and to the S-unit attacks listed in Section 1.2.

These computations have a long history before [118]: see, e.g., the systematic tables of number fields surveyed in [118, Appendix B]. Almost all of the number fields covered by these tables, and by the book, are non-cyclotomic. This does not mean that cyclotomic fields play such a minuscule part in the literature as a whole: on the contrary, there is an equally long history of algebraic number theorists exploring the special structure of cyclotomic fields (see, e.g., [352]) and exploiting this structure to efficiently carry out useful computations that nobody knows how to perform efficiently for general number fields.

An example of cyclotomic structure is the ability to instantaneously write down generators for the group of "cyclotomic units". These generators have two properties that are important for known attacks:

- The generators are *short*. See [72, Section 8] for quantification.

[7] Examples: [8] incorrectly claims that NTRU Prime's "abandonment" of cyclotomics was "motivated by recent progress in quantum algorithms for finding short vectors in principal ideal lattices with a guaranteed short generator"; and [20, "Algebraic attacks on cyclotomics?"] states a "Long history" incorrectly claiming that, *after* discussions in 2015 of [96], NTRU Prime's field was proposed "as a defense against certain conjectured advances in this line of quantum algebraic cryptanalysis".

- The group of cyclotomic units is a *finite-index* subgroup of the unit group. Computations show that the index is generally very small, making the full unit group easy to compute via the known technique of "saturation".

For example, the cyclotomic-unit lattice is the entire unit lattice for the 512th and 701st cyclotomic fields, and has index just 3 for the 1229th cyclotomic field. See [**352**, page 421].

Efficiently finding short generators of a finite-index subgroup of the unit group for *general* number fields is an unsolved problem. The sophisticated unit-group algorithms in, e.g., [**118**] have much worse than polynomial scaling. At first glance a quantum computer might seem to help, since the Eisenträger–Hallgren–Kitaev–Song quantum algorithm [**144**] computes unit groups in polynomial time, but the resulting generators are not short.

For number theorists, it is not surprising that the extensive literature on faster computations for cyclotomic fields is reflected in recent literature demonstrating faster attacks against various cryptographic problems using cyclotomic fields. This is not because other fields have been ignored; it is because cyclotomics provide tremendously helpful tools for the algorithm designer.

Some people will tell you that these speedups for cyclotomics are a reason to *use* cyclotomics, rather than a reason to *avoid* cyclotomics. This is like

- saying that for a stream cipher one should select RC4 rather than ChaCha20 since, as noted above, there are many years of papers doing more and more damage to RC4's security; or
- saying that within discrete-logarithm cryptography one should use small-characteristic multiplicative groups rather than large-characteristic elliptic-curve groups.

History shows how dangerous it is to assume that every insecure cryptographic system will be immediately identified as insecure. Sometimes a security failure is recognized only as the culmination of a long series of papers on attack speedups. Proactive risk management requires recognizing the attack speedups as an alarm bell, not treating these speedups as something to be embraced.

3 The risk table

Table 1.1 summarizes the risks in the NISTPQC lattice KEMs. This section defines each row in the table.

3.1. Basics, part 1: "pk+ct bytes" and "ct bytes". The "ct" row states the number of bytes used by a ciphertext, while the "pk+ct" row states the total number of bytes used by a public key and a ciphertext.

The motivation for considering "pk+ct" is that this is the data communicated if a key is sent to one user, used for one ciphertext, and then discarded. The motivation for considering "ct" is that this is the data communicated if a key is sent to many users through a lower-cost broadcast channel and then used for many ciphertexts.

For reasons to think that "ct" is more important than "pk+ct", see [258] and [50, Section 2.3]. It is, for example, not difficult for a server to broadcast keys through the Internet's Domain Name System, which has automatic local caching of data by Internet service providers. A client then retrieves keys locally, but still has to send a ciphertext all the way back to the server. See [51].

As noted in the table caption, each column takes the largest specified KEM size that fits pk+ct into 4KB, except that the `frodo` column allows 20KB. The "pk+ct" and "ct" rows for `frodo` are marked in red. The reason for this exception is that no specified `frodo` size fits into 4KB.

3.2. Basics, part 2: "errors" and "Q or P". This refers to the top-level classification of NTRU variants given in [69] and [49, Section 8].

A public key reveals a "multiplier" G and an approximation $A = aG+e$ to the multiple aG. The "Quotient" NTRU systems take $A = 0$, so G is the quotient $-e/a$. The "Product" NTRU systems instead take random G (or something that one hopes looks sufficiently random; see Section 3.16). A ciphertext reveals an approximation $B = Gb + d$ to Gb, and for the Product NTRU systems also reveals an approximation $C = Ab + M + c$ to $Ab + M$.

"Errors" refers to how the small secrets a, b, c, d, e are chosen. In all cases a, b are chosen randomly. "Noisy" NTRU means that c, d, e are also chosen randomly. "Rounded" NTRU obtains approximations, except for the $A = 0$ approximation in Quotient NTRU, by rounding: d is determined by rounding Gb to B; e for Product NTRU is determined by rounding aG to A; c for Product NTRU is determined by rounding $Ab + M$ to C.

3.3. Basics, part 3: "modulus", "dimension", first 2-norm", "second 2-norm". These are *some* quantitative features of the lattice problems that appear, meant solely to give a flavor of the variations. See [49, pages 48–52] for five tables stating the full problems for all round-2 lattice submissions. Beware that those do not all match the problems in round-3 lattice submissions.

The set of multipliers G can always be written in the form $((\mathbf{Z}/q)[x]/F)^{k\times k}$ for a monic polynomial $F \in \mathbf{Z}[x]$. "Modulus" refers to q, and "dimension" refers to $k \deg F$. For NTRU, some computations are carried out modulo $x^p - 1$ while others are carried out modulo $(x^p - 1)/(x - 1)$; for purposes of this document, F is $(x^p - 1)/(x - 1)$.

Security analyses of the difficulty of determining b, d from G and $B = Gb + d$ are influenced not just by the modulus and dimension but also by the size of the secrets b and d. Similar comments apply to other secrets. The secrets are sometimes drawn from two different distributions: e.g., an explicit distribution of b and an implicit distribution of d from rounding. "First 2-norm" and "second 2-norm" refer to the typical Euclidean norms of the two secrets, sorted into non-decreasing order.

The numbers in the table show that submissions vary in the balance that they choose between these parameters. Some submissions take somewhat larger dimensions, for example, while others take somewhat larger secrets. Quantitative attack improvements could change the optimal balance in either direction.

3.4. Basics, part 4: "$\log_2$(Core-SVP)", "ct/$\log_2$ ratio". "Core-SVP" is a particular mechanism of assigning a (pre-quantum) "security level" to each lattice system, specifically the "0.292β" column defined in [12]. The motivation for using Core-SVP is that NIST has made comments such as "we feel that the CoreSVP metric does indicate which lattice schemes are being more and less aggressive in setting their parameters" and has repeatedly criticized submissions that used other metrics.[8]

It is important to realize that Core-SVP is a combination of underestimates, overestimates, possible underestimates, and possible overestimates, making its relationship to the actual cost of attacks highly unclear—even within the limited scope of attacks that it considers, ignoring the fact that attacks are advancing. See [65, Section 6] for a detailed review of (1) the attacks covered, (2) how Core-SVP estimates the cost of those attacks, and (3) many open questions regarding the actual costs of those attacks.

The "ct/$\log_2$ ratio" column divides the number of ciphertext bytes by the $\log_2$ of Core-SVP. The motivation for computing this quotient is that sizes of optimized lattice systems are *roughly* linear in $\log_2$(Core-SVP).

The Core-SVP column is marked in red when it is below 2^{256}. The ratio column is marked in red when it is above 16, corresponding to Core-SVP 2^{256} requiring ciphertext size (without key size) above 4KB.

3.5. What qualifies as a "known attack avenue"? NIST has stated [271] that it is "open to the possibility" that there is an attack against cyclotomics, and that it is also "open to the possibility" that there is an attack against $x^p - x - 1$. Why, then, does Table 1.1 have a row for "cyclotomics" and not a row for "$x^p - x - 1$"? More generally, if there's a split of systems, some systems having feature X and some systems having feature Y, then shouldn't there be a row for the possibility of X being a problem, and a separate row for the possibility of Y being a problem?

To see that this is not the right rule, consider the following special case of the rule: an analysis of SARS-CoV-2 transmission risks via different systems for faculty meetings should consider

- the possibility of transmission via aerosols and
- the possibility of transmission via Zoom.

[8] However, NIST has not criticized the round-3 Kyber submission for switching to a different definition. The metric in [12] measures the LWE problem for KEMs that are based on LWE, Ring-LWE, and Module-LWE, while it measures the LWR problem for KEMs that are based on LWR, Ring-LWR, and Module-LWR. The new metric in [25, Table 4] measures a mixed LWE/LWR problem, even though the Kyber submission continues to state [25, Sections 1 and 4.4] that it is "based on the hardness of . . . MLWE" and that its security claims are "based on the cost estimates of the best known attacks against the MLWE problem underlying Kyber". For clarity, this document uses "Core-SVP" for the metric in [12] and "revised-Core-SVP" for the metric in [25]. The new round-3 version of Kyber-512 has Core-SVP 2^{112} and revised-Core-SVP 2^{118}.

The first possibility is a legitimate subject of risk analysis: there is an explanation of a *mechanism* by which SARS-CoV-2 could transmit via aerosols (see, e.g., [**351**]), and one can scientifically study this mechanism to understand the risk better. There is no explanation of any such mechanism for Zoom.

There has been a long history of attacks damaging the security of lattice-based cryptography; see, e.g., many examples listed in Section 1. The tools used in these attacks are mechanisms that could enable further attacks doing even more damage. The attack avenues listed in Table 1.1 are a categorization of known attack tools. Every tool known for attacking $x^p - x - 1$ also applies to cyclotomics, while the opposite is not true; this is why there is a "cyclotomic" row and not an "$x^p - x - 1$" row. (Some literature claims that cyclotomics are "uniquely protected" against a particular attack strategy, but this claim is false; see Section 5.6.)

As an example unrelated to cyclotomics, one could consider the *possibility* of "Noisy" being broken while "Rounded" survives, and the *possibility* of "Rounded" being broken while "Noisy" survives. (Theorems stating that a "Rounded" attack implies a "Noisy" attack are too weak to apply to these KEMs; see Section 5.3.) But what's the *mechanism* that threatens one and not the other? The lack of an answer to this question is why "Noisy" and "Rounded" are not listed as attack avenues separately from "lattices".

Similarly, one could consider the rank-2/3/4 module systems being broken while the ring-based systems survive, or vice versa. (Theorems stating that a ring attack implies a module attack are too weak to apply to these KEMs; see Section 5.3. Theorems stating that a module attack implies a ring attack are also too weak to apply to these KEMs; see Section 5.4.) But there's again no explanation of the *mechanism* that threatens one and not the other. This is why "rings" and "modules" are not listed as attack avenues separately from "structured lattices".

One can also generically view any attack that quantitatively benefits from lower dimensions as a reason to prefer higher dimensions, or generically view any attack that quantitatively benefits from smaller secrets as a reason to prefer larger secrets. However, these attacks do not have sharp cutoffs at any particular dimension or any particular size of secret, and there is no explanation of a mechanism that could create such cutoffs. (Some literature, notably [**16**], claims that hybrid combinatorial-lattice attacks are "induced by the fact that secrets are ternary", but this is another false claim; see [**55**].) This is why "lower dimension" and "smaller secret" are not listed as attack avenues separately from "structured lattices".

3.6. Implementation attacks are not included. Table 1.1 does not cover attacks that depend on the implementation being attacked: for example, timing attacks, and attacks exploiting bugs. This exclusion should *not* be taken as a statement that these attacks are unimportant. On the contrary:

- The timing attack from [**169**] was a disaster for the Frodo software. Other submissions, such as NTRU Prime, already have software written within frameworks that verify immunity to timing attacks; this is a clear advantage.

- Other NISTPQC software implementations have been broken because of incorrect generation of random objects. See, e.g., [**235**] and [**316**]. The use of rounding in NTRU Prime and in SABER eliminates some (not all!) of these random objects.
- Optimizations are a major driver of cryptographic software complexity and bugs. Many vectorized subroutines in the NTRU Prime software have already been computer-verified to match the reference subroutines on all inputs. See [**61**].

The reason for this exclusion is that merely *assessing* implementation risks is a major research project that has not been completed. The risks in Table 1.1 are thus limited to risks that are intrinsic in the KEM specifications, risks that cannot be addressed by changes in implementations.

One can argue that implementation risks are less important than specification risks in the long run. It is clearly feasible to eliminate timing attacks with current technology. One *hopes* that research into high-assurance cryptographic software will eliminate KEM bugs within a few years. It is more difficult to stop other side-channel attacks such as electromagnetic attacks (see Section 7.10), but one can still hope for safe deployments on devices kept far enough away from attackers.

3.7. Known attack avenue: "lattices". All of these KEMs are broken if one finds a sufficiently short nonzero vector in a given lattice. Note that "short" does not mean specifically "shortest". This row is for attack tools that seem to apply to all lattices. Half of the recent attacks listed in Section 1, such as the sieving advances, are of this type. The risk is that lattices in general continue to lose security.

3.8. Known attack avenue: "derandomization". This row marks systems where the underlying PKE is randomized: PKE decryption recovers only part of the randomness used to produce a ciphertext. As part of the "CCA transform" used to try to protect against chosen-ciphertext attacks, the KEM derandomizes the PKE, choosing the random coins in the PKE as deterministic functions of the PKE input.

Incorrectly ignoring proof looseness (see Section 5.8) leads to the unjustified conclusion that derandomization cannot lose security. Fixing this error shows that there is a *possibility* of losing security. This, by itself, would still not qualify derandomization for listing in Table 1.1. However, Bernstein [**59**] recently found examples of randomized PKEs that have attacks exploiting derandomization: the KEM is approximately q times easier to break than the PKE. Here q is the number of hash queries, around 2^{100} for a large-scale attacker today.

These attacks rely on a partial information leak from the PKEs. The core risk here is that algorithms can similarly extract partial information from lattice PKEs. This risk is structurally avoided by the KEMs using deterministic PKEs.

3.9. Known attack avenue: "decryption failures". This row marks KEMs where the underlying PKE does not always succeed in decrypting the ciphertexts that it produced. The number listed in the row is λ when decryption is claimed to fail with probability at most $1/2^\lambda$. The number is marked in red when λ is below $\log_2$(Core-SVP).

There is a common perception that decryption failures are not an issue if they are rare enough that *legitimate users* will not encounter them. However, as noted in [**49**, Section 5], attackers can search offline for decryption failures. The core algorithmic question is how reliably attackers can *recognize* decryption failures offline (based on, e.g., the magnitude and pattern of errors in each ciphertext), not sending most of the ciphertexts to the legitimate user. Proofs do not address this risk; see Section 5.7.

Some of the recent attacks listed in Section 1 are decryption-failure attacks, arising in some cases from λ having been calculated incorrectly (which is not listed here as a separate risk) and in some cases from the impact of decryption failures having been underestimated. The risk is that these attacks continue to advance. This risk is structurally avoided by systems without decryption failures.

3.10. Known attack avenue: "structured lattices". Recall that the set of multipliers G can always be written in the form $((\mathbf{Z}/q)[x]/F)^{k\times k}$ for a monic polynomial $F \in \mathbf{Z}[x]$. This row marks systems where $\deg F > 1$.

Section 1.2 listed various claimed "barriers" broken by S-unit attacks. These attacks exploit structured lattices. There is a risk that the latest "barriers" claimed between these attacks and Ring-LWE/Module-LWE will also fall. The general risk of structured lattices is highlighted in the Frodo submission [**15**, Section 1.2.1]:

> Given the unpredictable long-term outlook for algebraically structured lattices, and because any post-quantum standard should remain secure for decades into the future—including against new quantum attacks—we have based our proposal on the algebraically unstructured, plain LWE problem with conservative parameterizations (see Section 1.2.2).

The risk is structurally avoided by the Frodo design.

3.11. Known attack avenue: "cyclotomics". Within "structured lattices", this row marks lattices where the underlying number field is a cyclotomic field. In the case of SABER and Kyber, the field is defined by the cyclotomic polynomial $x^{256}+1$. In the case of NTRU, the field is defined by the cyclotomic polynomial $(x^p-1)/(x-1)$ where p is, e.g., 701 or 1229.

Cyclotomic structure is exploited in some of the number-theoretic attacks listed above; see Section 2.6. The risk is that cyclotomic structure enables further attacks. This risk is structurally avoided by Frodo, which does not use number fields in the first place, and by NTRU Prime, which uses non-cyclotomic fields.

3.12. Known attack avenue: "reducibility". This row is marked when $(\mathbf{Z}/q)[x]/F$ is not a field; in other words, when there are ring morphisms from $(\mathbf{Z}/q)[x]/F$ to smaller nonzero rings. There is, for example, a ring morphism from $(\mathbf{Z}/8192)[x]/((x^{701}-1)/(x-1))$ to $(\mathbf{Z}/2)[x]/((x^{701}-1)/(x-1))$, and there is a ring morphism from $(\mathbf{Z}/3329)[x]/(x^{256}+1)$ to $(\mathbf{Z}/3329)[x]/(x^2+17)$.

Such morphisms are exploited in the attacks of [**338**], [**145**], [**146**], [**105**], and [**106**]; the attacker applies the ring morphisms to public ring elements. Some of the attacks are on cases known to be breakable in other ways, but [**106**] breaks some prime-cyclotomic cases with larger error distributions than have been attacked in any other way. So far the attacks have not affected any NISTPQC KEMs, but the risk here is that the attacks are further extended. Note that this is not the same as the cyclotomic risk, although it is incurred by the same KEMs in Table 1.1.

"Modulus switching" relates some problems in $(\mathbf{Z}/q)[x]/F$ to problems in $(\mathbf{Z}/r)[x]/F$ for a replacement modulus r (although it adds considerable noise, as noted in [**69**, page 37]). Every polynomial F is reducible modulo various r, so attacks of this type are included under the general "structured lattices" risk.

3.13. Known attack avenue: "quotients". The problem of finding a, e given G and $A = aG + e$ could be easier in the homogeneous case that $A = 0$, the problem of recovering a, e given the quotient $G = -e/a$.

The analyses of [**206**] and [**139**] conclude that standard BKZ attacks for the homogeneous case do better than the standard analysis indicates if the modulus q is large, specifically above $0.004n^{2.484}$ where n is the dimension. Both papers also claimed that attacks of this type exist only for the homogeneous case,[9] but this was disproven by [**65**, Section 6.3]. The analysis of the known attacks against the inhomogeneous case is an open question; it could be that the standard analysis is wrong in all cases.

Given the lack of knowledge regarding the performance of known attacks, this row in Table 1.1 presumes that there is a security loss specific to the homogeneous case. Accordingly, this row marks the Quotient NTRU systems. The core risk is that the security loss can be pushed to smaller values of q, small enough to apply to NISTPQC submissions.

3.14. Known attack avenue: "extra samples". This row is marked for systems that release more than n "samples", in the terminology typically used for dimension-n LWE. These are exactly the Product NTRU KEMs.

For each error distribution, known attacks suddenly break all of the lattice problems considered here once enough samples are released to the attacker. See, e.g., [**134**], [**22**], and [**11**]. This row covers the risk of attack advances reducing the cutoff for the required number of samples.

One can also consider the risk that an attack already starts working at n samples, affecting all of the KEMs in the table. This is already covered by the general "lattices" risk and is not covered in this row. As an analogy, one can

[9] For example, [**206**, Section 7] claimed that an "important difference between NTRU and Ring-LWE instances is the fact that in NTRU lattices, there exists many short vectors". NIST [**7**] then stated a confused version of the same claim, namely that "the NTRU construction produces a lattice that has somewhat more structure than in similarly efficient RLWE and MLWE cryptosystems (due to having shorter than expected vectors)" and similarly that `sntrup` uses "a lattice with unusually short vectors". Note the difference between "many short vectors"—something that [**206**] emphasizes as critical—and "shorter than expected vectors"—which is simply false.

also consider ways that an improved attack against Quotient NTRU could be accompanied by an improved attack against Product NTRU; this is covered in the general “lattices” row, not the “quotients” row.

3.15. Known attack avenue: “non-QROM FO”. All of these KEMs use variants of the Fujisaki–Okamoto transformation [**153**] as CCA conversions. These variants use a hash function. It’s trivial to write down hash functions that eliminate all KEM security even if everything else is unbroken; the risk in this row is that the selected hash function also loses security.

There is a misconception that hash-function weaknesses are outside the scope of KEM evaluation as long as KEMs use standard hash functions. The problem here is that there is a gap between

- the security properties for which standard hash functions were reviewed (e.g., preimage resistance) and
- the security properties that these KEMs require from the hash functions (this is more difficult to write down—essentially, one has to state what security means for the specific KEM at hand).

This gap needs to be addressed by further security analysis. Claims that standard hash functions are “indistinguishable from random” are ill-defined and are not a substitute for security analysis.

3.16. Known attack avenue: “non-QROM 2”. The Product NTRU KEMs use, inside their PKEs, another hashing layer. The motivation for this layer is that a random multiplier G takes space in public keys, and having one standard G shared by everyone raises security concerns. These PKEs eliminate the sharing and almost all of the space issue by sending a short seed that is hashed to obtain a “random” G.

It’s again trivial to write down hash functions that eliminate all security here. The risk in this row is that the selected hash function loses security. This is analogous to the previous row but requires a different security analysis.

NIST [**8**, page 17] criticized two of the three options for this hashing layer in the Round5 submission. NIST stated that “there is no proof of security” for those options. However, as in Section 3.15, there is *also* no proof of security for the use of standard hash functions in this context. Each combination of hash function and KEM raises a different security question.

3.17. Known patent threats: “patent 9094189” and “patent 9246675”. Patent risks and attack risks involve different mechanisms, but patent risks are like attack risks in that they can easily trigger the worst-case scenario that the cryptographic user ends up being unprotected.

Another feature shared between patent risks and attack risks is that properly evaluating them requires expertise. For example, non-experts tend to think that a patent covers only what is literally included in the patent’s claims; but the “doctrine of equivalents” states [**346**, page 732] that a patent “is not limited to its literal terms but instead embraces all equivalents to the claims described”. The patent holder can show infringement of a patent claim under this doctrine by showing for each element of the claim that "the accused product performs substantially the same function in substantially the same way with substantially the same result" [**345**, page 1312].

U.S. patent 9094189, expiring 2032, is on essentially the LPR cryptosystem, and was filed before the LPR cryptosystem was published. It was filed after *unstructured* lattice systems were published, so it cannot cover those systems. Given the doctrine of equivalents and other procedures used by patent courts, all efficient LPR derivatives using *structured* lattices are threatened; there is no reason to think that one can escape the doctrine of equivalents by switching from Ring-LWE to Ring-LWR or by switching to low-rank modules. The "patent 9094189" row in the table is marked for the Product NTRU systems that use structured lattices.

U.S. patent 9246675, expiring 2033, is on a variant of the LPR cryptosystem with compressed ciphertexts, and was filed years before a 2014 paper [**302**] that claimed novelty for compressing LPR ciphertexts. Given court procedures, all LPR variants using shorter ciphertexts than LPR are threatened, even if the compression mechanisms are not identical. The "patent 9246675" row in the table is marked for the Product NTRU systems that use structured lattices and have smaller ciphertexts than LPR.

For unclear reasons, NIST has discouraged public analysis of NISTPQC patent threats; see, e.g., [**268**] and [**62**]. NIST has not released its own patent analysis. In response to a FOIA request regarding NISTPQC patents, NIST refused to provide 3 documents totaling 31 pages; see [**58**]. For years NIST also carried out patent-buyout negotiations; the details are secret and appear to have failed.[10]

There has nevertheless been some public analysis, including a paper [**241**] claiming "non-applicability" of the first patent "to Kyber and Saber", and an accompanying summary [**240**] claiming that Kyber/SABER patent risks "should be ignored". The paper fails to recognize the definitions and procedures used in patent courts, such as Markman hearings and the doctrine of equivalents.

There was also some litigation, starting in 2017, against the European version of the first patent. The first round of litigation failed to kill the patent. An appeal also failed to kill the patent, and ended on 20 October 2021. See [**147**] for the list of litigation documents. For comparison, [**310**] claimed in 2020 that the patent was "likely" to be invalidated by prior art.

3.18. Systemic risks: "PKE instability" and "instability". These rows look at the history of the *family* of KEMs, rather than any specific KEM size. The "PKE instability" date is the publication date of the most recent change

[10] See, e.g., [**93**] (another patent holder stating in June 2019 that "discussions started Friday"); [**62**] (quoting NIST email in December 2020 stating "We are working to clear up the IP situation, but it is a slow process"); [**58**] (quoting NIST document in January 2021 stating "We're still mostly waiting for them to give us a number"); and [**270**, video, minute 1:08-1:09] (NIST statement in June 2021 that "just the ballpark figures that were being mentioned made it clear that this wasn't something NIST was gonna be able to do. It was definitely more than our overall budget for our crypto group. We're continuing to talk to them").

in the family of underlying mathematical PKEs. The "instability" date is the publication date of the most recent change in the family of KEMs, including the PKEs, CCA conversions, and encoding details.

Instability contributes to attack risks because a moving target (1) adds to the overall reviewer load, (2) discourages researchers whose work on a previous version no longer applies to the current version, and (3) leaves a shorter period for analysis. "PKE instability" is separated from "instability" because most of the attack analysis has focused on the underlying PKEs.

Instability also contributes to general patent risks because a change could bring a system within scope of a patent filed in the meantime. Normally each patent application is published 18 months after filing (for the full picture see, e.g., [**252**]), so patent applications filed by April 2019 would normally have been online by October 2020; serious effort by NIST to organize labor-intensive searches and public analyses of patent applications might have completed within a year; but no such effort has been announced. Separating "PKE instability" from "instability" also makes sense in this context, since it allows separate analyses of PKE-related patents and CCA/encoding-related patents.

4 Conclusions from the risk table

Let's now use Table 1.1 to rationally select the least risky lattice KEM. Of course, if new events—for example, sudden success in patent buyouts—warrant updates in the table then the selection process should be revisited.

4.1. The size decision. Within each KEM family (for example, `ntruhps`), all available evidence is that the larger KEMs are harder to break. Perhaps a larger KEM also protects against future advances. One can construct arguments that a smaller KEM *might* do better; but a larger KEM is less risky. Consequently, the application should select the largest KEM that fits its performance constraints. The remaining question is simply which of the seven families to take.

4.2. The structured-lattice decision: `frodo` vs. everything else. The "ratio" row shows that `frodo` requires an order of magnitude more space to achieve any particular Core-SVP level than the other lattice KEMs. If the goal is to handle applications that require a small lattice system, as in the Google example from Section 1.4, then `frodo` simply isn't an option.

What happens if the importance of these applications has been overstated, and the applications of interest can actually afford 10KB ciphertexts?

These applications reach Core-SVP only 2^{150} with `frodo`, and much higher security margins with the other submissions. This includes `ntruhrss`, which has an `ntruhrss1373` option that wasn't listed in the table because the ciphertexts are 2401 bytes. The structured-lattice submissions could specify parameter sets at much higher Core-SVP levels using 5KB or even 10KB ciphertexts. A closer look suggests that this would be problematic for `kyber` without some tweaks to the `kyber` structure (see Appendix A.3), but higher Core-SVP levels would be straightforward for, e.g., `ntruhps`.

Perhaps much larger security margins from other KEMs will be destroyed by advances in structured-lattice attacks. For comparison, the $\mathbf{F}_{2^n}^*$ discrete-logarithm catastrophe mentioned in Section 2.3 was a quantitatively larger loss of security levels for that problem. On the other hand, selecting `frodo` would

- exacerbate the general lattice risks—half of the recent attack advances listed in Section 1—exactly because of the much lower security margin and
- incur a separate risk of losing all security if we're wrong about applications being able to afford 10KB.

Even if the devastating-structured-lattice-attack risk is assigned higher weight than the general-lattice-attack risk, it's hard to see how it can also be assigned higher weight than the application-can't-afford-this risk. This eliminates `frodo`.

Note that the decision-making process in this section is assuming that the goal is to select *one* of the lattice-KEM families. Selecting multiple families would reduce the application risk, although it would also increase systemic risks: for example, the risk of error increases when effort is spread across multiple families. On the other hand, this risk is lower for the two NTRU Prime families, `sntrup` and `ntrulpr`: there is extensive sharing of the work across these NTRU Prime options, notably in the parameter choices, security analysis, software, testing, and verification. Providing both options via NTRU Prime is simpler and less error-prone than gluing together two non-unified submissions. There is also some sharing of work between `ntruhps` and `ntruhrss`, and some of the software work has been shared between NTRU and NTRU Prime.

4.3. The Quotient-vs.-Product decision: `ntruhrss` and `ntruhps` vs. `saber` and `kyber`. These four systems share some risks. Beyond the shared risks, `ntruhrss` and `ntruhps` also incur the "quotients" risk, while `saber` and `kyber` also incur the "derandomization", "decryption failures", "extra samples", "non-QROM 2", "patent 9094189", and "patent 9246675" risks.

There is a myth that Product NTRU systems such as `saber` and `kyber` have been proven to be as secure as Quotient NTRU systems such as `ntruhrss` and `ntruhps`. Any such theorem would logically imply that the "derandomization", "decryption failures", "extra samples", and "non-QROM 2" risks cannot favor the Quotient NTRU systems. But there is no such theorem; the myth arises from an indefensible misrepresentation of known theorems. See Section 5.

There are generic arguments that attacks keep improving so the "quotients" risk should be taken seriously. For the same reasons, the "derandomization", "decryption failures", "extra samples", and "non-QROM 2" risks should be taken seriously. There are no obvious risk-management principles that would justify saying that the "quotients" risk outweighs the other four risks.

More importantly, the "quotients" risk does not outweigh the patent risks. As NIST put it in the call for NISTPQC submissions [279], it is "critical that this process leads to cryptographic standards that can be freely implemented in security technologies and products".[11] A patent threat doesn't need to be

[11] For comparison, [323] says "I fully trust that NISTPQC does exactly what it set out to do in 2016 and selects post-quantum cryptography algorithms primarily based

a guaranteed disaster to interfere with free implementability. This eliminates `saber` and `kyber`; it also eliminates `ntrulpr`, even though `ntrulpr` avoids some of the cryptanalytic risks.

4.4. The number-theoretic decision: `ntruhrss` and `ntruhps` vs. `sntrup`. Within the remaining three systems, there are four shared risks; `ntruhrss` and `ntruhps` also have the "cyclotomics" risk and the "reducibility" risk; `sntrup` has no extra risks. This eliminates `ntruhrss` and `ntruhps`, leaving `sntrup` as the least risky system.

This analysis did not account for the systemic risks of instability, but these are also in favor of `sntrup`. The PKE family inside `sntrup` was already published in 2016, years before the other submissions made changes to their PKE families. Looking beyond the PKE family at the entire KEM family shows changes in `sntrup` in 2019, but other submissions also changed their KEM families at that point. In short, `sntrup` provides the maximum stability of any of these KEM families.

4.5. Why did NIST downgrade NTRU Prime to alternate? NIST's procedures are not transparent, but a FOIA request led to the publication of [20], including the following statements:

> [bullet item:] Among the remaining, structured lattice schemes, our assessment was that cyclotomics (esp power-of-2 cyclotomics) are the clear "community standard"
> [bullet item:] So, we moved Kyber, Saber, NTRU on as Finalists, but kept NTRUprime too

NIST appears to have made its decision regarding NTRU Prime entirely upon this basis, never mind any of the other NTRU Prime features.

Procedurally, NIST appears to be requiring Kyber to be broken outright as a prerequisite for considering NTRU Prime, whereas it certainly isn't requiring NTRU Prime to be broken outright as a prerequisite for considering Kyber. There's extensive literature carrying out efficient computations for cyclotomics where the best known algorithms for most number fields are much slower; NIST doesn't treat this as an alarm bell regarding cyclotomics. This literature includes, much more recently, efficient breaks of famous cryptosystems using cyclotomics; NIST still doesn't treat this as an alarm bell regarding cyclotomics. Cyclotomic KEMs win by default because they're supposedly the "community standard".

Back in 2016, Google's NewHope experiment [224] triggered development and then submission of many NewHope variants to NISTPQC, all using cyclotomics by default. However, 41% of the round-1 lattice submissions provided options

---

on their technical and security merits (while of course also considering the advice from their own legal professionals.) This is what the security industry, other U.S. government departments, various military branches, international standardization bodies, etc expects it to do." The claim regarding NISTPQC is contradicted by [279], and the claim regarding expectations by "the security industry" etc. is contradicted by [256].

that do *not* use cyclotomics. This includes submissions from submission teams with many years of lattice publications, such as Frodo and Titanium. Most of these submissions paid heavily in performance for this choice. Most of these submissions expressed concerns regarding security. Most of these submissions provided *no* cyclotomic options. This is not the story of a community confident in the security of cyclotomics.

With this background in mind, let's think through what NIST's "community standard" statement from [**20**] is saying:

- The word "remaining" means that this statement isn't about, e.g., Titanium; NIST had already thrown Titanium away on the basis of performance.
- The word "structured" means that this statement isn't about Frodo.

The statement, by its own terms, is limited to a selection of just four submissions: NTRU, NTRU Prime, SABER, and Kyber. Yes, three of those use cyclotomics, but this is selection bias, not a "community standard". One could just as easily exclude any other targeted submission by selecting a distinguishing feature of that submission and claiming that the feature is not the "community standard". This is a completely unprincipled way to be making decisions. What we should all be doing instead is proactively minimizing risks.

5 No, theorems have not ruled out these risks

Let's look more closely at the pervasive advertising of the "provable security" of lattice KEMs. The main function of the advertisements is to make the reader believe that various lattice risks have been mathematically proven not to exist.

This section summarizes what has actually been proven regarding each topic. The main conclusion is that, because of important limitations in the theorems, the theorems *do not* rule out any of the risks listed in Table 1.1.

5.1. There is no proof that breaking `newhope1024` or `kyber768` is as hard as breaking `ntruhrss701`. Typical advertisement [**140**] making the reader think that there *is* a proof: "The preference for using Ring/Module-LWE is due to the fact that this problem is at least as hard as NTRU".

The reader is led to believe that KEMs based on Ring-LWE or Module-LWE, meaning Product NTRU KEMs such as `newhope` or `kyber`, have been proven to be at least as secure as KEMs based on NTRU, meaning Quotient NTRU KEMs such as `ntruhrss`. Perhaps there's an exception if the NTRU dimensions are larger than the Ring-LWE/Module-LWE dimensions, but surely the advertising means that `newhope1024` and `kyber1024` and `kyber768` have been proven to be at least as secure as `ntruhrss701`. This, in turn, means that there cannot be a security reason to take `ntruhrss701` rather than `kyber1024`; ergo, the extra risks of `kyber1024` in Table 1.1 cannot exist.

However, what was actually proven (in [**305**, page 33]) is vastly weaker than this; is useless for comparing the KEMs submitted to NISTPQC; and does not rule out any of the risks in Table 1.1. Section 5.2 looks more closely at the proof.

5.2. What the "Ring-LWE is at least as hard as NTRU" proof actually says. Say one has an algorithm A that *with high probability*, say probability $1-\epsilon$, solves the *search* Ring-LWE problem *with n samples* and error distribution χ for the ring $R = (\mathbf{Z}/q)[x]/F$, where F has degree n. This means that one can find χ-distributed secrets b, d given a uniform random $G \in R$ and $B = Gb + d$, with b, d, G being chosen independently.

Let's use this algorithm to solve the problem of *distinguishing* a quotient e/a from uniform, where e, a are χ'-distributed elements of R^*, the set of invertible elements of R. The distinguisher works as follows: choose b, d randomly from distribution χ, apply the algorithm A to inputs $G = e/a$ and $B = Gb + d$, and see whether A returns (b, d). Notice that $B = G(b + a) + d - e$, so *if* there's a big overlap between the distributions of (b, d) and $(b + a, d - e)$ then A can't tell whether it should be returning (b, d) or $(b + a, d - e)$ in the overlap, so it will frequently fail, say with probability $\geq \delta$, where one can calculate δ from seeing how small χ' is related to χ. (One can also consider $(b + ma, d - me)$ for other small ring elements m.) If, however, e/a is replaced by something uniform then, by assumption, A returns (b, d) with probability $1 - \epsilon$. *If* $\delta > \epsilon$ then this guarantees a distinguisher with probability at least $\delta - \epsilon$.

But what does this have to do with KEM security? The advertisement abuses naming to sweep this critical question under the rug. The reader is invited to confuse the first problem with the security of Ring-LWE-based KEMs, and to confuse the second problem with the security of NTRU-based KEMs. Instead of skipping the question, let's look at how little these problems have to do with the security of, say, `newhope1024` and a hypothetical `ntru1024` using the same ring $R = (\mathbf{Z}/12289)[x]/(x^{1024} + 1)$.

Starting from the distribution χ in `newhope1024` and the distribution χ' in `ntru1024`, the proof calculates some $\delta < 1$; the exact value of δ won't matter for the following analysis. The proof concludes that a solution to the *search* Ring-LWE problem *with n samples* and with success probability $1 - \epsilon > 1 - \delta$ implies a *distinguisher* between e/a and uniform with distinguishing probability at least $\delta - \epsilon$.

But where's the proof that a distinguisher for e/a, even a high-probability distinguisher, breaks Quotient NTRU? Sure, e/a is (modulo irrelevant details) the public key; so what? One of the useful features of Quotient NTRU is that the KEM ROM IND-CCA2 security analyses boil down to the simple question of whether the underlying PKE is *one-way*, meaning that a random plaintext is hard to find from a ciphertext and public key; one doesn't have to analyze the more subtle problems of ciphertext distinguishers or public-key distinguishers. See [49, Sections 6 and 7].

Furthermore, the proof that "Ring-LWE is at least as hard as NTRU" does *not* convert any of the following disaster possibilities for NewHope, Kyber, etc. into a distinguisher:

- There could be an n-sample search Ring-LWE attack that succeeds with probability $1 - \epsilon$ for some $\epsilon \geq \delta$. The proof says nothing in this case. This wouldn't be an issue if δ were *extremely* close to 1, but the `newhope1024` distribution isn't wide enough to achieve that; also, Kyber uses narrower distributions than NewHope does, very close to NTRU's traditional ternary distributions.

- There could be a high-probability search Ring-LWE attack exploiting the extra samples released by NewHope, Kyber, etc. This is covered by the "extra samples" risk in Table 1.1. The proof says nothing about this—it assumes specifically an n-sample attack.
- There could be a KEM attack that uses a distinguisher for the underlying PKE to exploit the derandomization in NewHope, Kyber, etc. This is covered by the "derandomization" risk in Table 1.1, a risk that doesn't apply to Quotient NTRU; see also Section 5.8 below. The proof says nothing about this—it assumes a *search* Ring-LWE attack.
- There could be a KEM attack finding outputs of encapsulation that don't decrypt correctly in NewHope, Kyber, etc. This is the "decryption failures" risk in Table 1.1. The proof says nothing about this.
- There could be a loss of security from the hash function that's actually used to compute G in NewHope, Kyber, etc. This is the "QROM 2" risk in Table 1.1. The proof says nothing about this.

Structurally, the proof hypothesis considers just one type of attack, ignoring all of the above attack avenues. Meanwhile the proof conclusion isn't a successful attack against the Quotient NTRU KEMs. The advertising nevertheless leads readers to believe that the proof says that any attack against NewHope or Kyber implies an attack against Quotient NTRU KEMs.

5.3. Weaponization of selective proof exaggeration. If it is acceptable to disregard proof limitations then one can argue exactly the opposite position—namely, that one should prefer Quotient NTRU since Quotient NTRU has been proven to be at least as hard as Ring-LWE—by saying that the Stehlé–Steinfeld cryptosystem [**341**] is a Quotient NTRU cryptosystem having a proof that an attack implies a Ring-LWE attack. However, the limitations of *this* proof have been prominently advertised: most importantly, the e, a distributions in that cryptosystem are much bigger than in the Quotient NTRU KEMs in NISTPQC.

One can similarly argue that one should prefer rounding to noise as follows: disregard the proof limitations of [**32**, Theorem 3.2], claim that LWR has been proven to be at least as hard as LWE, claim that Ring-LWR has been proven to be at least as hard as Ring-LWE, etc. (Even better for fans of rounding, there don't seem to be any proofs pointing in the opposite direction.) But this proof is another case where the limitations have been prominently advertised.[12]

Why, within lattice-based cryptography, does common proof advertising

[12] As an example of limitations being ignored in *some* logically inapplicable proofs, consider [**20**], which claims that "security proofs generally assume Gaussian noise" as a reason for expressing concern regarding the security of fixed-weight noise, but does not point to the LWR-vs.-LWE proofs as a reason for expressing concern regarding the security of LWE. On the contrary, [**20**] says "No known reduction from LWE to LWR, but no known attacks on normal parameter ranges for LWR KEMs", so the reader is thinking about whether there are unknown attacks against LWR. Why

- exaggerate most of the proofs considered in this section, such as the "Ring-LWE is at least as hard as NTRU" proof reviewed above, but
- highlight the limitations of other proofs, such as the proofs from [**341**] and [**32**]?

Perhaps this inconsistency is related to the fact that the proof exaggerations all make Kyber look better, while ignoring the limitations of [**341**] and [**32**] would make Kyber look worse. In any case, whatever the intent might be, exaggeration of security proofs damages the risk-assessment process, and defenses need to be put into place to protect against this damage.

5.4. There is no proof that breaking `kyber1024` is as hard as breaking `newhope1024`. Here [**227**] says that "Mod-SIVP can trivially be shown to be no easier than Id-SIVP" and that "M-LWE and M-SIS are obviously no easier than R-LWE and R-SIS". These refer to proofs that one can solve various dimension-1024 lattice problems for the ring $R = (\mathbf{Z}/q)[x]/(x^{1024}+1)$ if one can solve, e.g., dimension-4096 lattice problems for the module R^4. But if another cryptosystem uses the module S^4 over the *vastly smaller* ring $S = (\mathbf{Z}/q)[x]/(x^{256}+1)$, then these proofs say merely that an S^4 attack implies an S attack—which

- is useless, since S is too small to be secure, and
- does not say that an S^4 attack implies an R attack.

Meanwhile proofs pointing in the opposite direction, such as "large modulus Ring-LWE $\geq$ Module-LWE", have the "large modulus" limitation prominently displayed and are (correctly) not treated as reasons to prefer NewHope to Kyber.

A different proof works around the S-vs.-R size gap as follows: S is isomorphic to the subring $(\mathbf{Z}/q)[x^4]/(x^{1024}+1)$ of R. Then R is an S-module isomorphic to S^4, and multiplication by an element of R is a restricted case of multiplication by a 4×4 matrix over S. However, [**56**] lists reasons that this still doesn't give a proof that `kyber1024` is as hard to break as `newhope1024`. For example, the `newhope1024` error distribution is much wider, adding up 16 bits modulo 12289 where `kyber1024` adds up 4 bits modulo 3329. "Modulus switching" is too noisy to compensate for this.

5.5. There is no proof that breaking `kyber1024` is as hard as breaking `ntrulpr953`. One can object to the `newhope1024` comparison in Section 5.4 as a distraction from the comparisons that remain in NISTPQC. Surely what the proofs must mean is that module KEMs are at least as secure as ring KEMs *if* the error rate is at least as large, the dimension is at least as large, etc.: for example, the proofs must say that the module KEM `kyber1024` is as hard to break as as the ring KEM `ntrulpr953`.

But, no, there's no hope of any such proof, even if the moduli are equalized. The proof mentioned above, exploiting a subring of $(\mathbf{Z}/q)[x]/(x^{1024}+1)$ that is isomorphic to $(\mathbf{Z}/q)[x]/(x^{256}+1)$, obviously doesn't apply to the `ntrulpr953` ring $(\mathbf{Z}/q)[x]/(x^{953}-x-1)$, which is chosen to be a field.

---

does [**20**] not say "No known reduction from LWR to LWE, but no known attacks on normal parameter ranges for LWE KEMs"?

For a number theorist, the subrings of $(\mathbf{Z}/q)[x]/(x^{1024}+1)$ are interesting attack tools to explore. As noted in Section 2.6, there is already a long literature on related computations exploiting these tools. It would be unsurprising for $(\mathbf{Z}/q)[x]/(x^{1024}+1)$ to turn out to be weaker than $(\mathbf{Z}/q)[x]/(x^{953}-x-1)$; and there certainly isn't a proof to the contrary.

As a direct result of this gap, a full proof that attacks against a rank-4 $(\mathbf{Z}/q)[x]/(x^{256}+1)$ KEM imply attacks against a $(\mathbf{Z}/q)[x]/(x^{1024}+1)$ KEM wouldn't be able to draw any conclusions regarding a $(\mathbf{Z}/q)[x]/(x^{953}-x-1)$ KEM. Furthermore, the structure enabling such a proof, namely the relatively small ring $(\mathbf{Z}/q)[x]/(x^{256}+1)$ (with even smaller subrings), raises a variety of security questions that the proof is incapable of addressing. This is reminiscent of the situation described in [**44**]: "The structures used in the 'proofs of security', such as automorphisms, are also some of the structures exploited in this attack."

Is it clear at this point that these attack tools *will* damage the security of `kyber1024`? No. But readers should not be deceived into believing that the use of modules in `kyber1024` has been proven to not lose security.

5.6. Cyclotomics are not uniquely protected against reductions. The typical advertising here [**303**, page 15] claims that "cyclotomic fields, used for Ring-LWE, are uniquely protected" against a particular class of attacks. The reader is led to believe that this attack avenue threatens x^p-x-1 and that it does *not* threaten cyclotomics, contradicting what Section 3.5 said and implying that Table 1.1 should have an x^p-x-1 row.

However, checking the proof shows that the proof is limited to saying that cyclotomic fields are protected against certain attacks. The proof never showed that cyclotomic fields are *unique* in being protected. It is straightforward to see that the NTRU Prime fields are protected too.

The class of attacks is as follows:

- Starting from the polynomial F, find a modulus q and a root α of F modulo q where α has very low order. As a concrete illustration, let's assume α has order 4.
- Consider the ring morphism $(\mathbf{Z}/q)[x]/F \to \mathbf{Z}/q$ that maps x to α. This maps the powers of x to the powers of α, namely ± 1 and $\pm\alpha$ since α has order 4, so small linear combinations of powers of x map to small linear combinations of 1 and α.
- Apply this map to G and $B = Gb + d$. The objective of the attack is to distinguish B from uniform. (One can similarly consider more samples.)
- The usual distributions of b and d are small combinations of powers of x, so b and d map to small linear combinations of 1 and α, constraining the relationship between G and B. If q is sufficiently large then this distinguishes B from uniform.

One can convert this into a distinguishing attack for other moduli by "modulus switching", although this adds noise, requiring q to be larger.

This attack fails against the polynomial $F = x^p - x - 1$ used in NTRU Prime. Indeed, if $\alpha \in \mathbf{Z}/q$ has order 4 then it is a root of x^2+1. If it is also a root of

$x^p - x - 1$ then q must divide the resultant of $x^p - x - 1$ and $x^2 + 1$ in $\mathbf{Z}[x]$. This resultant is either 1 or 5, depending on p mod 4, so $q \leq 5$, which is far too small for the attack to work: the images of b, d are very well distributed across the integers modulo 5. The attack similarly fails against power-of-2 cyclotomics, since the resultant is 4.

The above calculations were only for order 4, but replacing x^2+1 with further cyclotomic polynomials shows that the attack fails against all of the NTRU Prime fields for the entire range of orders of interest. These calculations are explained in [**52**]. So, no, cyclotomics are not *uniquely* protected against this class of attacks. It is also interesting to note that attacks in this class *do* work against prime cyclotomics in some cases; see Section 3.12.

See also [**52**] for a survey of known extensions to this class of attacks. The extended class of attacks has a huge parameter space, and it is not clear how to find optimal parameters; there is no proof that cyclotomics—or any other fields—are immune to the extended attacks. There is also no proof ruling out further attacks. The risks here are covered in Section 3.12.

5.7. There is no proof that the decryption failures in `kyber1024` are safe. Typical advertisement [**25**, Section 4.3.1] on this topic: "Tight reduction from MLWE in the ROM ... The following concrete security statement proves Kyber.CCAKEM's IND-CCA2-security ... the security bound is tight". The reader understands this to mean that the KEM has been proven to reach its target ROM IND-CCA2 security level, under the stated assumption regarding the underlying mathematical problem, in this case Module-LWE.

This, however, has *not* been proven for the KEMs where "decryption failures" is listed in red in Table 1.1. See [**49**, Section 5] for a detailed analysis of what has been proven.

For example, for `kyber1024`, the ROM proof allows an attacker to succeed with probability $q/2^{172}$ where q is the number of hash computations carried out by the attacker. The proof thus does not guarantee any security against an attacker carrying out 2^{172} computations. This is a very large number of computations, but it is not the claimed 2^{256} security level.

A useful way to understand the risk here comes from the proof modularization in [**80**]. For the PKEs under discussion, it is reasonable to expect that plaintexts do not collide under encryption, so there *exists* an oracle that, unlike the actual decryption algorithm, always decrypts correctly. If the attacker cannot find a failing ciphertext then, for this attacker, the actual decryption algorithm behaves the same way as the oracle. The risk is that a ROM attack *can* find a failing ciphertext. Similar comments, with quantitative differences, apply to QROM attacks; see [**80**]. Non-QROM attacks are covered separately in Table 1.1.

5.8. There is no proof that derandomization in `kyber1024` is safe. The derandomization risk is the most deeply hidden risk considered in this section: the risk where the first attacks (against *any* KEMs, never mind lattice KEMs) were developed most recently, and the risk that takes the most work to see from the previous literature. General awareness of the risk is low enough that there has been little motivation for advertisements specifically mentioning this risk. However, more general advertisements such as

- "Ring-LWE is at least as hard as NTRU" (as mentioned above) and
- "there is no security ... reason to prefer an NTRU NewHope [Quotient NTRU using the NewHope ring] to NewHope" [**240**]

lead readers to believe that Product NTRU KEMs are provably as hard to break as Quotient NTRU KEMs. This belief forces, among other things, denial of the "derandomization" row in Table 1.1: Quotient NTRU KEMs do not have a derandomization risk, so the belief logically implies the belief that Product NTRU KEMs do not have a derandomization risk. Given these advertisements, it seems necessary to emphasize that this specific belief is incorrect.

For the Quotient NTRU KEMs under consideration, the underlying PKEs are deterministic: decapsulation naturally recovers all randomness that was used to generate a ciphertext. Deterministic PKEs have a proof that any ROM IND-CCA2 attack against the most popular KEM construction implies an attack with essentially the same speed and success probability against the one-wayness of the PKE. For a careful proof, see [**73**], which also presents counterexamples to some previously claimed "theorems" regarding other PKEs.

The Product NTRU KEMs do *not* have a proof that a ROM IND-CCA2 attack implies an attack with essentially the same speed and success probability against the one-wayness of the underlying PKE. What has been proven is more limited:

- There is a proof that loses a factor q in success probability, where q is the number of hash computations carried out by the attacker. This limitation is important: a current large-scale attack can reach $q \approx 2^{100}$, and NISTPQC KEMs are supposed to look ahead to a future of even larger attacks.
- There is a tight proof under the assumption that the PKE provides IND-CPA security, indistinguishability against chosen-plaintext attacks. This change of assumption is important: algorithm designers study IND-CPA attacks less often than they study one-wayness, the occasional studies have found much faster attacks in some cases, and there are more IND-CPA attack avenues than one-wayness attack avenues. See generally [**49**, Section 6].
- There is a tight proof starting from the newly formulated assumption that the PKE provides "q-OW-CPA security". This is implied by IND-CPA security, and perhaps it is satisfied even when IND-CPA is not, but it has received essentially no study from algorithm designers. The new attacks from [**59**] show that q-OW-CPA security can be approximately q times easier to break than one-wayness.

The underlying issue is that the original PKE is unable to recover all randomness that was used to generate a ciphertext. These KEMs begin by derandomizing the underlying PKE to obtain a deterministic PKE—but there is no proof that derandomization preserves security; [**59**] gives examples where it loses security.

5.9. There is no proof that breaking `kyber1024` is as hard as breaking worst-case Ideal-SVP. Common advertising: "Attractive features of lattice cryptography include . . . security under worst-case intractability assumptions" [**305**]; in particular, there is a "very strong hardness guarantee" [**238**] for Ring-LWE, assuming the hardness of worst-case Ideal-SVP.

The reader, easily putting everything together, is led to believe that breaking `kyber1024` is provably as hard as breaking the underlying Module-LWE problem for $((\mathbf{Z}/q)[x]/(x^{256}+1))^4$, which in turn is provably as hard as breaking a Ring-LWE problem for $(\mathbf{Z}/q)[x]/(x^{1024}+1)$, which in turn is provably as hard as breaking a worst-case Ideal-SVP problem.[13]

See [**72**, Appendix A] for examples of how the last step in this chain, the Ideal-SVP guarantee, has been highlighted in a series of proposals of lattice KEMs leading into NISTPQC. The Ideal-SVP guarantee plays several roles in influencing decision-making processes: in promoting the notion that changing moduli is guaranteed safe,[14] for example, and in discrediting Quotient NTRU.[15] (Meanwhile, for some reason, the importance of Ideal-SVP is downplayed in response to advances in attacks against Ideal-SVP.[16])

A closer look shows that, in fact, these proofs *do not* apply to `kyber1024` or any of the other KEMs under consideration for deployment, even if one considers only the underlying problems such as Ring-LWE. What the theorems say, in a nutshell, is that one can extract a *vastly less efficient* worst-case Ideal-SVP attack from an attack against a related Ring-LWE problem. The efficiency gap is polynomially bounded but so large as to make the theorems logically inapplicable to NISTPQC. See generally [**49**].

This issue was buried for many years inside theorem statements that said "polynomial" without saying what the polynomial was. Applying such theorems to any specific size was never logically justified: the theorem statements did not rule out the possibility of a gigantic polynomial. Micciancio–Regev [**259**] characterized taking the polynomial into account as "overly conservative" but did not indicate how large the polynomial was.

---

[13] Sometimes the chain of exaggerations extends another step, for example with the wording of [**308**]: "the underlying worst-case problems—e.g., approximating short vectors in lattices—have been deeply studied by some of the great mathematicians and computer scientists going back at least to Gauss, and appear to be very hard."

[14] For example, [**236**] stated that "we have a line of research that states that avoiding a modulus q that supports NTT is (at least asymptotically) unnecessary—the worst-case to average-case reductions don't care about the ring modulus (assuming that worst-case lattice problems are actually hard)".

[15] For example, NIST [**8**] stated that NTRU "lacks a formal worst-case-to-average-case reduction".

[16] For example, in [**21**], NIST claimed that there was a "barrier" between Ideal-SVP and Ring-LWE, as one of two reasons for judging that [**60**] "does not impact the ongoing standardization process as-is". (The other reason relied on the incorrect conclusions of [**137**]; see Section 1.2.) If Ideal-SVP attacks don't matter, then why is it a negative feature for NTRU to supposedly not have an Ideal-SVP guarantee?

The shocking magnitude of the polynomial in these proofs finally came to light when Chatterjee–Koblitz–Menezes–Sarkar [**104**] "analyzed Regev's worst-case-to-average-case reduction for a cryptosystem that Regev had proposed, took lattice dimension 1024 with security target 2^{128} as a case study, and found an astonishing 2^{504} tightness gap in the proof", in the words of [**49**, Section 9]. The more recent proof in [**159**] has a somewhat smaller polynomial but is similarly inapplicable to NISTPQC.

6 The importance of proper benchmarking

Performance problems can create security risks when they lead users to reduce security levels or to disable cryptography entirely. A thorough risk analysis needs to include a performance analysis. Frodo's much lower Core-SVP level at each size, compared to the other KEMs, already appeared in Table 1.1 and played a decisive role in Section 4.2, but smaller performance differences among the other KEMs could also influence risks.

This section and Section 7 study performance in much more detail. Given the conclusion of Section 4 that `sntrup` is the least risky lattice KEM, Section 7 looks in particular at the NTRU Prime performance numbers. This section looks at the general question of how to evaluate costs.

There are many traps and pitfalls in cost evaluation, as illustrated by [**212**] analyzing many different types of "benchmarking crimes".[17] The cryptographic case studies in this section illustrate that careful attention to benchmarking procedures is required to protect evaluations and comparisons against errors and sabotage.

6.1. Do benchmarks reflect the performance that the cryptographic user will see? Many years ago, "optimal extension fields" in elliptic-curve cryptography were advertised as allowing "faster multiplication and much faster inversion", in the words of [**42**]. However, a closer look at performance in [**42**] showed that these extension fields damaged overall performance:[18]

- The big inversion speedup didn't make much difference in context, since inversions weren't actually a serious bottleneck.
- These fields "have a huge disadvantage: even if they are slightly faster on some CPUs, they are much slower on other CPUs".

Presenting inversion microbenchmarks as if these represented total costs is an example of "benchmarking crime" B1 from [**212**], "microbenchmarks representing overall performance". Optimizing a design for one platform, and then presenting

[17] One can criticize the word "crime" as implying *mens rea* that is not necessarily present, even when the source obviously benefits from the crime. What matters for this document is the damage that these "crimes" do to comparisons, whether or not the damage was intentional.

[18] This is separate from the point made in the same paper that prime fields "have the virtue of minimizing the number of security concerns" for ECC.

results for that platform as if these represented performance across platforms, is an example of "benchmarking crime" A3, "selective data set hiding deficiencies".

Unfortunately, the same "benchmarking crimes" are a recurring problem in NISTPQC. Microbenchmarks selected to favor specific submissions are often presented as if they were overall performance, while other benchmarks showing deficiencies are suppressed.

6.2. Case study: `ntruhrss` vs. `kyber`. Consider recent claims that `ntruhrss` is "15X behind in speed" [240] compared to a lattice system proposed in 2021, which in turn was portrayed in [140] as having similar speed to "Ring/Module-LWE" systems. Readers can easily fall into the trap of believing that `ntruhrss` is 15× less efficient than `kyber`.

More broadly, readers can easily fall into the trap of believing that `ntruhrss` is poorly designed—for example, that it ignores "the huge research efforts that have been made in this area in the last 25 years" [240] and could easily be replaced by "a (much) better" variant in "6 months"—and therefore shouldn't be standardized. Such perceptions are influential: consider, e.g., [82] expressing concern that standardizing NTRU would be standardizing a "less-optimal" KEM.

But are the claims correct? Let's check encapsulation speed on an ARM Cortex-M4 microcontroller. This is not an obscure scenario: NIST said [19] that microcontroller comparisons should use ARM Cortex-M4, and common sense suggests that people sending encrypted messages from such lightweight platforms are more likely to encounter speed bottlenecks than people sending encrypted messages from Intel chips. The benchmarks of M4-optimized software collected by the ongoing `pqm4` project [201] show

- `kyber512` taking 551681 cycles,
- `kyber512-90s` taking 445609 cycles, and
- `ntruhrss701`, with higher Core-SVP, taking 375948 cycles.

This looks like a solid win for NTRU over Kyber. How is this possible, if NTRU is "15X behind"?

6.3. Does key-generation time matter? Let's look more closely at where the "15X behind" claim comes from. The underlying paper [140, Table 3] shows the following Intel Skylake cycle counts:

- "319.9K" for "NTRU-HRSS-701",
- "20.4K" for the new "NTTRU",
- "23.3K" for "Kyber-512", and
- "38.9K" for "Kyber-768".

The ratio between 319.9K and 20.4K is 15.7; more to the point, the ratio between 319.9K and 23.3K is 13.7.

All of these are totals of keygen+enc+dec time. The 319.9K is, according to [140, Table 3], dominated by keygen time. But wait a minute: where's the evidence that the user cares about keygen time in any of these systems? There is a long history of user annoyance at the latency of *RSA key generation*, but this does not justify drawing any conclusions regarding much faster lattice KEMs.

All of these KEMs are designed so that keys can be freely reused. Eliminating key-generation time immediately drops the `ntruhrss701` cost below 100K cycles. Declaring that keys have to be ephemeral has negligible effect on this speedup: a dual-core 2GHz Skylake laptop generating a new `ntruhrss701` key each minute is spending under $1/1000000$ of its cycles on key generation.

Note that *ephemeral* and *one-time* are different concepts. The security goal of "forward secrecy" means *promptly* erasing secret keys, so that an attacker later stealing the device doesn't have a copy of those keys. This is exactly what *ephemeral* keys—keys "lasting for a very short time"—accomplish. Having keys be *one-time* is neither necessary nor sufficient for this. Compared to a protocol properly designed to erase each secret within a specified number of minutes (see, e.g., [**48**]), a protocol using one-time keys

- is less efficient—because it generates a new key for each ciphertext—and
- fails to achieve the security goal—see, e.g., [**223**].

Erroneously equating one-time keys with ephemeral keys plays a prominent role in arguments to assign a high weight to key-generation time.

Furthermore, even *if* an application really wants one-time keys, inversions can be made much faster via "Montgomery's trick". See [**66**], which estimates a $2\times$ speedup for `ntruhrss701` keygen and demonstrates a much larger speedup for `sntrup761` keygen—including demonstrating web browsing on top of TLS 1.3 using `sntrup761` with fast key generation.

All of these points are covered in the literature: see, e.g., [**65**, Section 5.3, "Does key-generation time matter?"]. All of these points are ignored in the presentation of "319.9K" as overall performance.

6.4. Are benchmarks reproducible and properly labeled? Anyone who checks the SUPERCOP benchmarks [**71**] for Skylake (`samba`) sees that

- `kyber512` takes 37336 cycles for enc and 28629 cycles for dec, total 65965 cycles, while
- `ntruhrss701` takes 25740 cycles for enc and 61530 cycles for dec, total 87270 cycles.

These numbers show 32% more cycles for `ntruhrss701` than for `kyber512`, but `ntruhrss701` also has 15% higher $\log_2$(Core-SVP) than `kyber512` does. Also, `ntruhrss701` wins in pure enc time, and it is easy to argue that this is what matters if the sender (for example, a server doing enc) is busy while the receiver (for example, a laptop doing dec) is not.

For comparison, [**140**, Table 3] reports "34.6K" cycles for "NTRU-HRSS-701" enc and "65K" cycles for dec. The total of these cycle counts is 14% higher than what one sees for `ntruhrss701` in SUPERCOP. Why is a claim of a speedup over `ntruhrss701` comparing to `ntruhrss701` timings that are slower than previously published, publicly verifiable timings for `ntruhrss701` on the same platform?

A more striking gap, $4\times$ rather than 14%, is between

- [**140**, Table 3] claiming that "Kyber-512" takes "23.3K" cycles for the total of keygen+enc+dec and

- SUPERCOP showing 89657 cycles for `kyber512`: 23692 for keygen, and the 37336 and 28629 mentioned above for enc and dec.

This 4× gap comes from the fact that the "Kyber-512" in [**140**, Table 3] is *not* the Kyber-512 KEM submitted to NISTPQC. One might guess that it's the Kyber-512-90s KEM—but, no, it isn't that KEM either. Compared to Kyber-512-90s, this KEM is further modified to perform less hashing. To avoid confusion, let's refer to this KEM as Kyber-512-90s-PartialHash.

The Kyber presentation at the Third PQC Standardization Conference [**26**, video, 1:23:00, text on slide] advertised Kyber as being "conservative" in four ways. These four ways included

- "all symmetric crypto based on Keccak"—which is exactly what the "90s" version of Kyber abandons—and
- "shared key depends on full transcript"—which is abandoned by the further modification from 90s to 90s-PartialHash.

The mislabeling of Kyber-512-90s-PartialHash as "Kyber-512" in [**140**] will make many readers believe that Kyber-512, *with* its advertised features, achieves the speeds displayed in [**140**]; in fact, it is 4× slower.

Now that we've seen the actual Kyber performance figures, let's look again at the idea that, since `ntruhrss701` is supposedly "15X behind" the new "NTTRU", NTRU is poorly designed and shouldn't be standardized. Even if we accept the questionable idea that Skylake keygen+enc+dec is the sole metric of interest, a more reasonable estimate is 9×, given the keygen speedups from [**66**] and the faster enc+dec in SUPERCOP. Meanwhile Kyber-512 is 4× slower in this metric than the new Kyber-512-90s-PartialHash. Does this 4× slowdown mean that Kyber-512 was poorly designed and shouldn't be standardized?

The Kyber submission [**25**, Section 2] tells a very different story:

> As also mentioned in Section 1.5, hashing the public key and the ciphertext is not required for CCA security, but is instead done to make the function more robust and directly usable in applications that require the shared key to depend on the entire transcript. Our rationale is that because the basic operations comprising Kyber are extremely fast, we can afford to pay a time penalty and make the default version of Kyber as robust and misuse-resilient as possible.

One can similarly find the NTRU submission stating design goals other than performance. So, no, the fact that new designs use many fewer cycles doesn't mean that these submissions were poorly designed.

6.5. Does faster on some CPUs mean faster across platforms? There's extensive literature studying performance variations across platforms, including many examples where speedups on one platform are slowdowns on another. Chip designers make different decisions regarding which packages of bit operations to bundle into CPU "instructions" (and into "coprocessors", and into FPGA "slices", and so on); these decisions interact in many ways with algorithm optimization.

Kyber was designed from the outset to streamline a particular NTT-based multiplication strategy for Intel CPUs. The Kyber presentation at the round-3 NISTPQC conference claimed [26, video, 1:22–1:23] that

> the fastest way to perform arithmetic in the polynomial rings you find in, well, structured lattice-based cryptography is to use NTT-based multiplication. Now, by now I think all candidates use NTT-based arithmetic but Kyber was built to make this particularly efficient, which, well, makes it particularly efficient.

These blanket statements were promptly contradicted by the next presentation at the same conference: SABER does use NTTs on some platforms, but on other platforms it (1) is implemented without NTTs and (2) achieves speeds that Kyber still has not matched and perhaps cannot match. Perhaps most importantly, all available evidence is that Kyber hardware is less efficient than SABER hardware, primarily because Kyber uses more bit operations than SABER for arithmetic.

What about NTRU? The literature does not seem to contain the necessary bit-operation analysis, but the best bet from current knowledge is that—outside key generation—NTRU will use even fewer bit operations than SABER.

The main arithmetic bottleneck in NTRU is multiplying elements of, e.g., the ring $(\mathbf{Z}/8192)[x]/(x^n - 1)$. One element is small, having n coefficients in $\{-1, 0, 1\}$, while the other looks random, having n 13-bit coefficients. Schoolbook polynomial multiplication performs n^2 multiplications of 13-bit coefficients by small coefficients and accumulates the results; one is free at any moment to replace x^n with 1 and to replace 8192 with 0. For a full analysis one needs to not just count bit operations in this process but also see what speedups are possible with fancier multiplication methods: Karatsuba, Toom, and NTTs. See generally [41].

In SABER, an n-coefficient ring is replaced by (e.g.) 3 elements of an $n/3$-coefficient ring, which one has to multiply by a 3×3 matrix. The $3\times$ smaller ring makes schoolbook multiplication $9\times$ faster, but there are 9 multiplications—no obvious difference in speed, and no obvious reason that fancier multiplication methods would favor SABER. However, the small coefficients in SABER are larger, requiring considerably more bit operations for each coefficient operation. Furthermore, the high-level structure of Product NTRU, used in the SABER submission, uses more arithmetic for enc+dec than the high-level structure of Quotient NTRU, used in the NTRU submission. Operations beyond arithmetic, such as hashing, do not obviously favor SABER.

If one thinks of a multiplication of two 16-bit integers—this means hundreds of bit operations—as one "instruction" then it becomes difficult to see the bit-operation advantage of smaller inputs. The NTTs in Kyber take advantage of these instructions. However, people building ASICs, cryptographic coprocessors, etc. see the intrinsic hardware cost of each operation, and can build much more efficient hardware that takes advantage of how small the inputs are. SABER does better than Kyber on ASICs by using non-NTT multipliers (see [320]), and one would expect an NTRU ASIC to do even better. There are no good choices for a Kyber ASIC:

- An ASIC using NTTs is bottlenecked by full-size multiplications, which are much more expensive than multiplications with small inputs.
- An ASIC trying to avoid using NTTs faces serious problems since various objects communicated by Kyber are in the NTT domain.

Kyber's cross-platform difficulties go beyond ASICs. Consider the paper [**13**] targeting a "commercially available smart card chip (SLE 78)" with an RSA coprocessor, stating that Kyber "requires the usage of the Number Theoretic Transform (NTT), which we cannot realise efficiently with our approach", and achieving much better speeds by introducing a new design—labeled as a "Kyber variant" but actually much closer to SABER. The same approach works even more efficiently for NTRU and NTRU Prime, not requiring any modifications.

In short, Kyber's NTT-related design decisions look good in *some* benchmarks but look bad in others. It is critical to take a wide enough range of benchmarks to be able to see the full performance picture.

6.6. The dominance of communication costs for lattice KEMs. The total costs of a cryptographic system go beyond the costs of computation, such as the cost of encapsulation; they also include the costs of communication, such as the cost of sending the ciphertext produced by encapsulation.

These costs are naturally measured in different units, such as cycles and bytes. How does one put these on the same scale? One answer—used in [**65**, Section 5.4] and the earlier documents cited there—is to consider a quad-core 3GHz CPU handling a 100Mbps Internet connection. In one millisecond, the CPU runs 12 million cycles, while the Internet connection transmits only 12500 bytes. Let's model the CPU and the Internet connection as having similar costs; more precisely, let's model each byte on the Internet connection as having the same cost as 1000 cycles on the CPU.

This model provides a way to quantify the idea that communication costs are much more important for lattice KEMs than they are for ECC:

- If an ECDH system uses 200000 cycles and sends 64 bytes, the total cost in this model is 264000 cycles, about 3/4 from computation. If another system saves half the bytes, 32 bytes, the total cost drops to 232000 cycles, 88% of the first cost. If a third system then saves half the cycles, the total cost drops to 132000 cycles, 56% of the second cost.
- If a lattice system uses 100000 cycles and sends 1000 bytes, the total cost in this model is 1100000 cycles, about 90% from communication. If another system saves half the cycles, the total cost is 1050000 cycles, 95% of what it was a moment ago. Because so much of the cost is from communication, the cycle-count improvements make far less of a difference than they did in the ECDH example.

As a real example, let's consider `ntruhps4096821` (Core-SVP 2^{179}). The sender spends 44119 cycles on enc, and sends a 1230-byte ciphertext, costing 1230000 cycles; the total cost is 1274119 cycles, 97% from communication. The receiver receives the 1230-byte ciphertext, costing 1230000 cycles, and spends

81114 cycles on dec; the total cost is 1311114 cycles, 94% from communication. *If* the costs of a new key are included then there's a more noticeable 414070 cycles for keygen, but also another 1230 bytes to send the key and another 1230 bytes to receive the key; the total cost across both sides is 5459303 cycles, with 90% from communication. Making keygen $2\times$ faster might sound dramatic if keygen microbenchmarks are considered in isolation, but saves only 4% of the total cost.

It is hard to see how emphasizing the cycle counts of `ntruhps4096821`, rather than the sizes, can be justified in this context.[19] Even if the costs of a new key are included, saying that 539303 cycles are *as expensive as* 4920 bytes means saying that a quad-core 3GHz CPU is as expensive as a 1Gbps Internet connection. This seems hard to explain, given that a $500 computer with a quad-core 3GHz CPU will typically last for several years while Google Fiber 1Gbps will accumulate $500 in charges within several months.

See [**50**, Section 2.3] for more evidence backing the idea that "communication volume is by far the most serious performance problem with lattice systems". For example, the Google–Cloudflare experiments with lattice systems (see [**224**] and [**217**]), including a variant of `ntruhrss701`, have consistently seen much more impact of bytes transmitted than of cycles. See also [**168**, page 6], which describes a smartcard with a "100 MHz" Cortex-M3 CPU and "$<$ 100 kB/s" communication—i.e., communicating a byte takes more time than 1000 cycles.

6.7. Performance loss #256. Consider the following competition between NTRU and Kyber. What's the smallest ciphertext size provided by the KEMs specified in each submission, subject to requiring Core-SVP to be at least 2^{128}?

Answer: NTRU's `ntruhps2048677` fits into 931 bytes. Kyber's `kyber768` is noticeably worse: 1088 bytes, 17% larger. This is another solid win for NTRU over Kyber.

If one changes from requiring Core-SVP 2^{128} to requiring Core-SVP 2^{118}, exactly the Core-SVP level[20] of `kyber512`, then suddenly Kyber wins instead: `kyber512` fits into 768 bytes, 18% smaller than `ntruhps2048677`. But why should these comparisons be carried out at the levels that Kyber happened to select?

One might think that, well, NTRU happened to pick some examples of sizes, and Kyber happened to pick some examples of sizes, and if there's a request for an intermediate size then NTRU and Kyber can simply specify intermediate sizes meeting the request. This is true for NTRU, but it's *not* true for Kyber.

Structurally, Kyber uses a rank-k module over $(\mathbf{Z}/q)[x]/(x^{256}+1)$, so the dimension is $256k$. There are no Kyber options between `kyber512` and `kyber768` and `kyber1024`: those are $k = 2$ and $k = 3$ and $k = 4$. If `kyber512`'s Core-

[19] One could, as noted in [**65**, Section 5.4], artificially reduce the communication-vs.-computation ratio by pairing "an expensive radio inside a recent smartphone" with "a cheap microcontroller manufactured on a 90nm fab". It was proposed in [**65**] to have the community "agree upon a spectrum of data points regarding the costs of communication and the costs of computation in various environments".

[20] This gives `kyber512` the benefit of switching from Core-SVP to revised-Core-SVP.

SVP 2^{118} isn't acceptable[21] then one has to jump all the way up to `kyber768`, a remarkable 42% increase in ciphertext size. SABER has the same problem. The NTRU structure is different, supporting a much more tightly packed list of dimensions.

See Figure 7.3 below for a comparison across all security levels. After the 42% jump from `kyber512` to `kyber768` in the volume of data communicated, there is another 44% jump from `kyber768` to `kyber1024`. One can draw a superficial analogy to the jumps in key sizes between, e.g., AES-128, AES-192, and AES-256, but (1) AES keys are much smaller than Kyber ciphertexts, (2) AES keys are rarely sent over the wire, and (3) setting some AES key bits to constants allows smooth tradeoffs between key size and security, whereas the only way that Kyber can save ciphertext size is by jumping far down in security level.

The importance of this performance issue in lattice-based cryptography is not a new observation. This issue was, for example, motivation #1 stated for Lyubashevsky–Peikert–Regev [**239**], a paper that generalized previous work for power-of-2 cyclotomics to provide an arbitrary-cyclotomic "toolkit for Ring-LWE cryptography":

> While power-of-two cyclotomic rings are very convenient to use, there are several reasons why it is essential to consider other cyclotomics as well. The most obvious, practical reason is that powers of two are sparsely distributed, and the desired concrete security level for an application may call for a ring dimension much smaller than the next-largest power of two. So restricting to powers of two could lead to key sizes and runtimes that are at least twice as large as necessary.

Kyber's 768/512 is just 1.5 rather than 2, but this doesn't eliminate the problem, as illustrated by the 17% Kyber loss described above.

6.8. Consequences of performance loss #256 for NISTPQC. The call for NISTPQC submissions states that each submission should—if possible—"specify *several* parameter sets that allow the selection of a range of possible security/performance tradeoffs" (emphasis added). NIST proposed five broad security "categories" (see Appendix B) and said that submitters could specify even more than five parameter sets to demonstrate flexibility:

> Submitters may also provide more than one parameter set in the same category, in order to demonstrate how parameters can be tuned to offer better performance or higher security margins.

The evaluation criteria (see Appendix A) state "security" as the top evaluation factor, then "cost", then "algorithm and implementation characteristics". The third-factor criteria say that

> Assuming good overall security and performance, schemes with greater flexibility will meet the needs of more users than less flexible schemes, and therefore, are preferable

[21] This again gives `kyber512` the benefit of switching from Core-SVP to revised-Core-SVP.

and list “It is straightforward to customize the scheme’s parameters to meet a range of security targets and performance goals” as an example of “flexibility”.

Consequently, according to the NISTPQC rules, NTRU’s amply documented flexibility to provide a full spectrum of dimensions is a positive feature, compared to Kyber’s enforced leaps from dimension 512 to dimension 768 to dimension 1024. The size competition in Section 6.7 illustrates how this flexibility lets NTRU reach combinations of security targets and performance goals that Kyber cannot reach.

The same comments apply to SABER: Kyber and SABER share the design decision to require modules over a 256-coefficient ring, a big step backwards from the traditional flexibility of NTRU. Note also that these performance losses turn into quantitative security losses if the application has, e.g., ciphertext-size limits that don’t happen to match what’s favorable to Kyber and SABER.

The problem is even more extreme for NewHope, which leaps from dimension 512 to dimension 1024. NIST’s round-2 report [8] pointed out Kyber’s advantage in exactly this respect over NewHope, as part of explaining why NIST eliminated NewHope:

> KYBER naturally supports a category 3 security strength parameter set, whereas NewHope does not.

This is also an advantage for NTRU over Kyber: NTRU “naturally supports” more “categories” than Kyber does. Also recall that the call for submissions valued “better performance or higher security margins” even within “the same category”.

Surprisingly, the same NIST report [8] fails to recognize this advantage of NTRU over Kyber, and even issues a blanket statement that NTRU is less efficient than Kyber:

> While NTRU is very efficient, it is not quite at the level of the highest-performing lattice schemes . . . NTRU has a small performance gap in comparison to KYBER and SABER . . .

See [53] for a detailed analysis of how a pro-Kyber “discretization attack” would limit comparisons to security levels favorable to Kyber, producing the incorrect conclusion that Kyber consistently outperforms NTRU. In fact, the performance winner between Kyber and NTRU depends on the application requirements.

7 Performance of NTRU Prime

The NTRU Prime design process has a clear priority order. First, within the design space of lattice systems, eliminate unstructured lattices, because the goal is to handle applications that *require* something small. Second, *within small lattice systems*, eliminate unnecessary complications in security review, because these complications exacerbate lattice security risks. Third, *within the systems that remain*, optimize the tradeoff between

- size and
- security against known attacks.

One would guess that this produces much worse performance than submissions that give performance optimization a higher priority, balancing performance with other concerns. A key insight—this is not something obvious; it is the result of detailed performance analysis—is that this guess is incorrect. NTRU Prime reduces the attack surface at surprisingly low cost. Sometimes NTRU Prime outperforms all of the other submissions.

7.1. A simple example where NTRU Prime has the best performance. As in Section 6.7, let's ask for the smallest ciphertext size provided by the KEMs specified in each lattice submission, subject to requiring Core-SVP to be at least 2^{128}, but now let's look at all five submissions, not just NTRU and Kyber. NTRU Prime is the winner:

- NTRU Prime (`sntrup653`): 897 bytes.
- NTRU (`ntruhps2048677`): 931 bytes.
- Kyber (`kyber768`): 1088 bytes.
- SABER (`saber`): 1088 bytes.
- Frodo (`frodo640`): 9720 bytes.

Obviously NTRU isn't far behind. Kyber and SABER are noticeably worse, with 21% larger ciphertexts than NTRU Prime, as a direct result of the module-related performance loss explained in Section 6.7.

7.2. Competitive size-security tradeoffs. Let's vary the example from Section 7.1 by asking, for each b, what ciphertext sizes are available when Core-SVP is required to be at least 2^b. The results are plotted in Figure 7.3. This is an update of [**50**, Figure 3.5] to account for

- extra parameter sets specified for NTRU,
- extra parameter sets specified for NTRU Prime,
- a modified version of `kyber512`,[22] and
- the SABER submission's announcement that, because of an error in [**12**], pre-round-3 versions of the SABER submission had reported incorrect (too high) Core-SVP results.

See [**50**, Section 2.3] for reasons to use size metrics, specifically ciphertext-size metrics; and see [**50**, Section 3] for visualization principles.

Each b corresponds to a horizontal line, and better submissions are farther left on that line. The smallest ciphertexts are achieved by `sntrup` for $118 < b \leq 129$, `ntruhps` for $129 < b \leq 145$, `sntrup` for $145 < b \leq 153$, `saber`/`kyber` for $153 < b \leq 181$, `saber` for $181 < b \leq 189$, `sntrup` for $189 < b \leq 209$, `saber` for $209 < b \leq 260$, etc. Both `sntrup` and `ntruhps` can easily improve their position in the graph by adding further parameter sets, whereas `kyber` and `saber` cannot; see Section 6.7.

[22] This figure, and the subsequent figures in this section, permits `kyber512` to substitute revised-Core-SVP for Core-SVP.

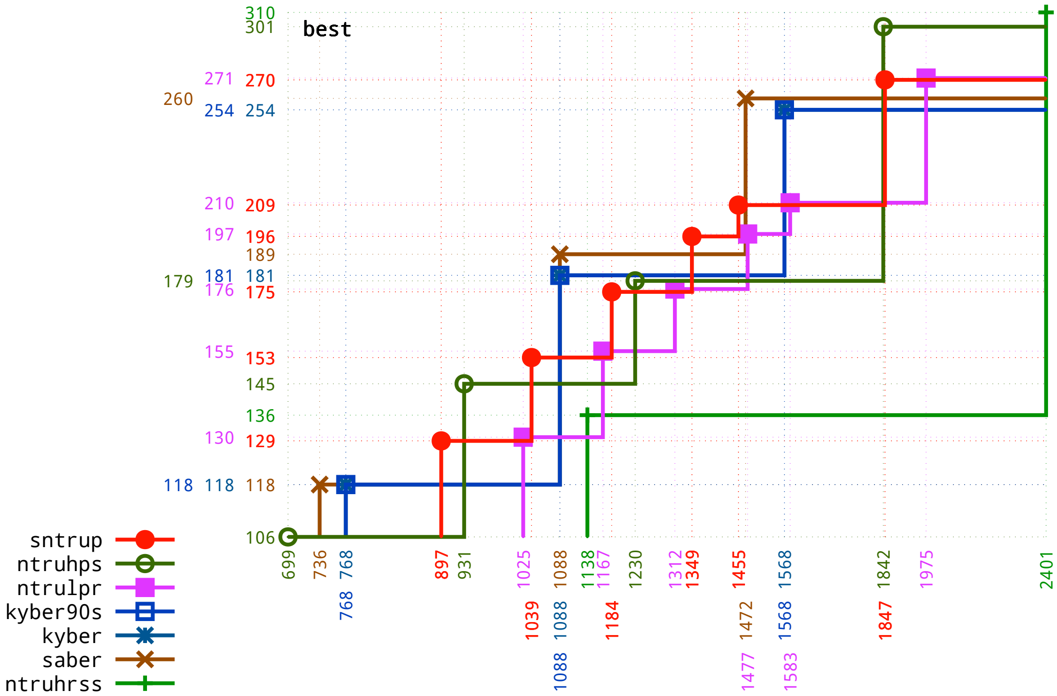

Fig. 7.3. Core-SVP (vertical axis, log scale) vs. ciphertext size (horizontal axis, bytes, log scale). Better security-performance tradeoffs are farther up and to the left.

One can also use the same graph to see how size constraints limit Core-SVP levels, exacerbating security risks. Each size limit corresponds to a vertical line, and better submissions are farther up on that line. For example, if an application is limited to 1024 bytes then it obtains Core-SVP 2^{145} with NTRU (`ntruhps`), Core-SVP 2^{129} with NTRU Prime (`sntrup`), and bleeding-edge Core-SVP with `saber` and `kyber`. Other size limits obtain the best Core-SVP with `saber`, in some cases tied with `kyber`, so this is not a clear argument against any particular KEM except for `frodo`.

Beware that lumping security levels into "categories" would (1) hide most of the information in Figure 7.3 and (2) let attackers manipulate the category boundaries so as to favor particular submissions. See [**53**] and Appendix B.

7.4. Competitive speeds. Section 6.6 explained the rationale for considering size, specifically ciphertext size, as the primary performance metric. But what if you have a slow ARM Cortex-M4 microcontroller—again, the ARM Cortex-M4 is NIST's designated microcontroller for comparisons [**19**]—and you really want to know how many cycles are used for computation?

One scenario is that the microcontroller is encrypting messages, typically to send to a larger device. Figure 7.5 shows the time for encapsulation. This graph

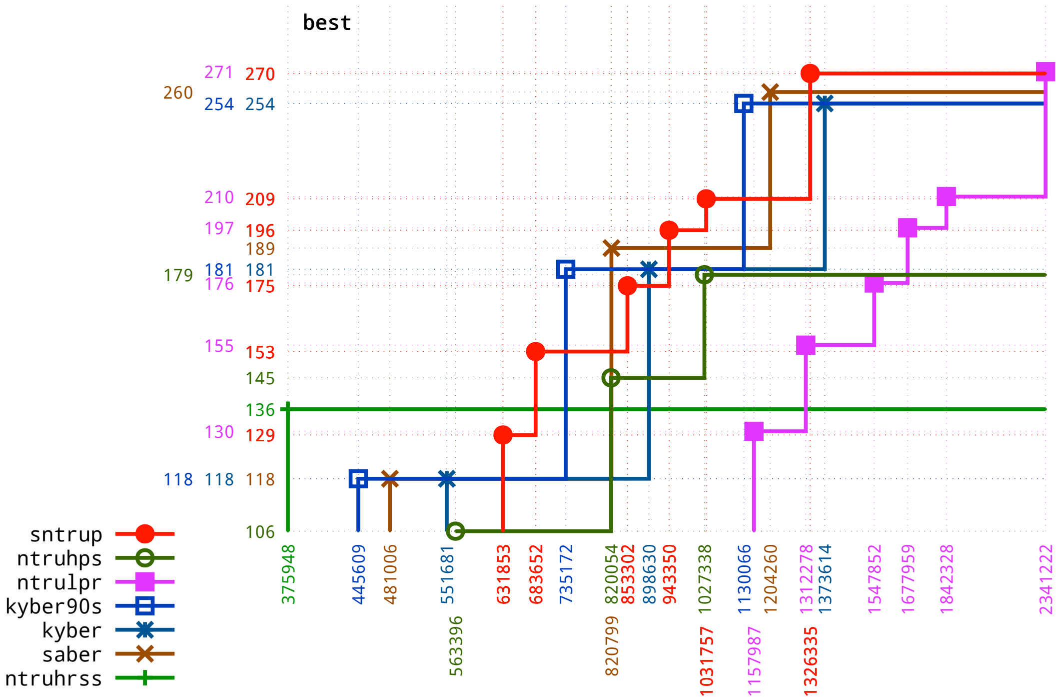

Fig. 7.5. Core-SVP (vertical axis, log scale) vs. ARM Cortex-M4 time for encapsulation (horizontal axis, cycles, log scale). Better security-performance tradeoffs are farther up and to the left.

shows that `sntrup` is competitive, including various security requirements (e.g., Core-SVP 2^{128}) for which it is the most efficient option.

Another scenario is that the microcontroller is decrypting messages. Figure 7.6 shows the time for decapsulation. In this graph, `sntrup` is the most efficient option at every security level—except for Core-SVP 2^{118} (and below), which, for security reasons, the NTRU Prime submission refuses to support.

All of these measurements are average cycle counts reported by the `pqm4` benchmarking project as of 17 October 2021 for M4-optimized software. The new `ntruhrss1373` and `ntruhps40961229` are not included since they do not yet have M4-optimized software.

Beware that neither graph includes the costs of communicating the ciphertext being handled. Including such costs would narrow the gaps between the systems, as in Figure 7.3, and would reduce `sntrup`'s overall advantage, but in any case `sntrup` is an attractive option for microcontrollers.

7.7. How is it possible for `sntrup` to be outperforming `kyber`? As noted above, NTRU Prime has a rule of not allowing performance improvements that would complicate security review. Kyber has no such rule: it emphasized speed from the outset and built a narrative of being the most efficient lattice KEM.

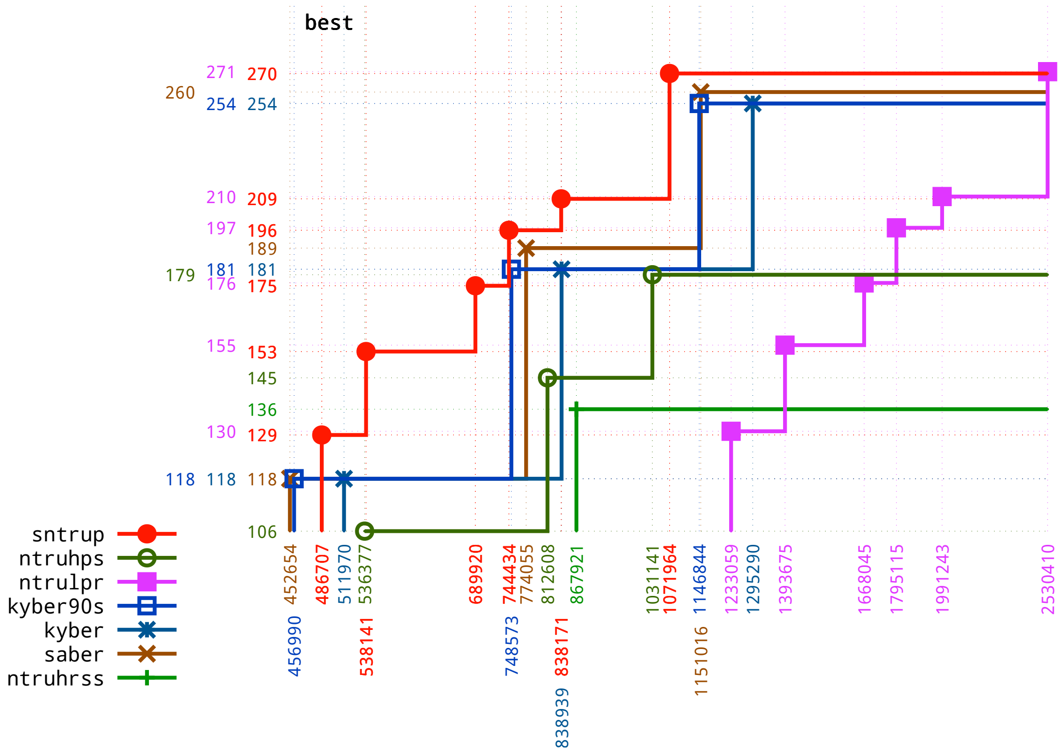

Fig. 7.6. Core-SVP (vertical axis, log scale) vs. ARM Cortex-M4 time for decapsulation (horizontal axis, cycles, log scale). Better security-performance tradeoffs are farther up and to the left.

However, Figures 7.3, 7.5, and 7.6 show `sntrup` outperforming `kyber` at many security levels in size, in encapsulation speed, and in decapsulation speed.

These numbers show that `sntrup` is doing something positive for performance that `kyber` failed to do. There are actually several differences contributing to `sntrup`'s performance wins. It's helpful to look at the details, to understand why `sntrup` can be expected to provide competitive cross-platform performance.

First, `kyber` is structurally limited to dimensions 256, 512, etc., while `sntrup` provides many more choices. See Section 6.7. The losses are easily visible from the big jumps of `kyber90s`/`kyber` costs in each of the figures; same for `saber`. On the other hand, this obviously can't be the whole story, given how consistently `sntrup` wins in Figure 7.6.

Second, as noted in [**65**, Section 5.5], all NTRU Prime operations take time $b^{1+o(1)}$ where b is the number of key bits. This is not true for `kyber`: the design decision to use modules over a 256-coefficient ring forces `kyber` to spend time handling $256k^2$ coefficients for lattice dimension $256k$. The cost per coefficient is small on Intel CPUs, but the cross-platform slowdown is more significant.

Note that modules do nothing to eliminate complications in security review: on the contrary, they *add* complications to security review. See [**65**, Section

4.8]. Consequently, modules have never been candidates for the NTRU Prime design. Fortunately for NTRU Prime's performance, modules also don't help performance. Regarding the idea that modules make optimized implementations easier, see Appendix A.3.

Third, by selecting a Product NTRU structure rather than a Quotient NTRU structure, Kyber sacrificed enc speed and dec speed in favor of keygen speed. From a performance perspective, this is making a highly questionable bet; see Section 6.3 and Section 7.11. Of course, if one simply declares that the metric of interest is keygen+enc+dec, rather than investigating the needs of applications, then the bet looks good.

For comparison, the NTRU Prime submissions in every round of NISTPQC have equally supported Product NTRU and Quotient NTRU. (This starts from the fact that each option avoids some security-review complications created by the other option. See, e.g., [**59**], and see generally Table 1.1.) From a performance perspective, under the reasonable assumption that enc speed and dec speed are much more important than keygen speed, NTRU Prime benefits from the Quotient NTRU option, `sntrup`.

Fourth, the way that Kyber bakes NTTs into its specification effectively ties implementors to a particular NTT-based multiplication strategy. This works well on Intel CPUs but turns out to damage cross-platform performance. See Section 6.5.

Fifth, even though the polynomial $x^p - x - 1$ used in NTRU Prime is slightly larger (in relevant metrics) than competing polynomials such as $x^p - 1$, the resulting performance loss turns out to be small, not dominating the performance picture. Similar comments apply to eliminating decryption failures.

7.8. Secret-key storage. Consider again a microcontroller using a KEM to decrypt messages. If the microcontroller is cycling through N secret KEM keys for, e.g., `kyber768`, then it needs to store $2400N$ bytes, since each KEM key uses 2400 bytes. This can be a problem, depending on

- how big N is,
- the total storage available in the microcontroller, and
- what else is competing for space in the microcontroller.

Figure 7.9 shows the number of bytes per secret key for all of the KEMs. Kyber and Saber consume the most space of all of the small KEMs; `ntrulpr` and `ntruhps` consume the least; `sntrup` and `ntruhrss` are in the middle.

A closer look at the KEM designs shows that each design has some natural flexibility regarding how much data is precomputed during key generation and stored in the secret key, rather than being recomputed during decapsulation. KEM designers make judgment calls regarding the best tradeoffs here. None of the KEMs opted to store only a short seed and regenerate keys from that seed during decapsulation; and none of the KEMs selected the opposite extreme, such as precomputing all possible transforms of multiplication inputs.

Secret-key holders can vary this decision while retaining interoperability, at the expense of a more complicated implementation ecosystem. A full graph of

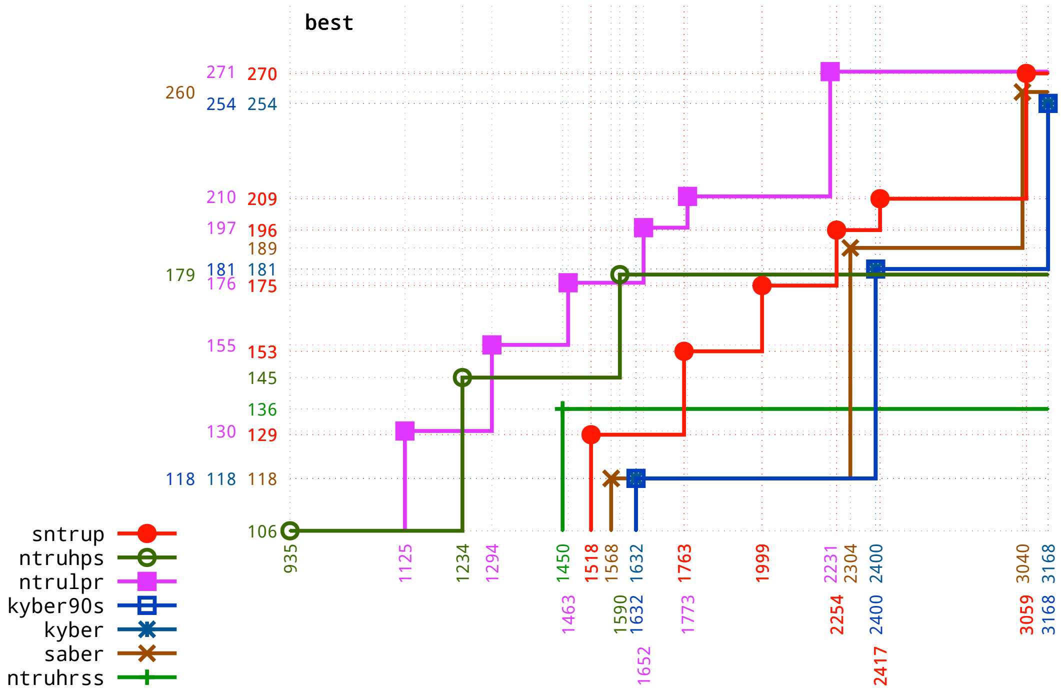

Fig. 7.9. Core-SVP (vertical axis, log scale) vs. secret-key size (horizontal axis, bytes, log scale). Better security-performance tradeoffs are farther up and to the left.

all of these options would require merging Figures 7.6 and 7.9 into a three-dimensional picture showing the tradeoffs between size, decapsulation cycles, and security, after implementation work to measure those tradeoffs.

Given that Kyber, with its own selection of secret-key size, is consistently outperformed by `sntrup` in Figure 7.6 (decapsulation time) *and* in Figure 7.9 (secret-key size), it is reasonable to guess that implementing more options for both Kyber and `sntrup` will show a three-dimensional picture favorable to `sntrup`.

Beware that the extra dimension here makes it even more important than usual to be on alert regarding improper benchmarking. This is illustrated by the range of benchmarking errors that appeared in response to [**63**], a recent message correctly disputing the idea "that Kyber is the most efficient lattice KEM in NISTPQC". Concretely, [**63**] observed that "if I want an ARM Cortex-M4 to decrypt messages, specifically with Core-SVP $\geq 2^{128}$" then Kyber uses 50% more cycles than NTRU Prime, according to `pqm4`, and also receives 20% more bytes than NTRU Prime. The conclusion of [**63**] was as follows:

> There are other benchmarks where Kyber does better, but cherry-picking pro-Kyber benchmarks and pretending that Kyber always wins is highly

inappropriate. The actual situation is that the performance winner depends on the application.

There were three directions of responses. First, there were various responses saying that Kyber does better on other benchmarks—shifting to Kyber's favorite security levels, highlighting key-generation time, and ignoring communication costs. Structurally, these responses were not addressing the point of dispute.

Second, there were objections to the RAM consumption and code size of the `sntrup` software. See [**23**] and [**322**] ("clearly not practical for use in embedded engineering"). A closer look shows, however, that

- the reported code-size numbers were dominated by *key generation*, which has no relevance to the scenario under discussion; and, furthermore,
- memory usage wasn't an optimization target for this code (see Section 7.10).

Experience suggests that NTRU Prime code fitting into much less memory will not lose much speed. As an analogy, compare the Saber results in [**116**], fast but not space-optimized, to the followup Saber results in [**2**], space-optimized without much loss of speed.

Third, there were responses such as [**237**] saying that Kyber can save time in decapsulation by spending more RAM per secret key. Quantitatively, increasing Kyber's secret key in this scenario from 2400 bytes to 5856 bytes was estimated to reduce Kyber from 839000 cycles to 445000 cycles and the "90s" version from 749000 cycles to 431000 cycles, while NTRU Prime uses 486707 cycles. The conclusion of [**237**] is that this makes Kyber "faster" than NTRU Prime. This conclusion ignores all of the following facts:

- In a reasonable scenario of a 100MHz CPU with 100kBps communication (see Section 6.6 and [**168**]), Kyber is still slower: it would need to decapsulate in 295000 cycles to catch up.
- $5856N$ bytes of RAM for N secret keys for Kyber (assuming keys are stored in RAM) are much larger than $1518N$ bytes for the competition. To put these numbers into perspective, consider [**23**] highlighting "16,088 bytes of RAM" used by (not RAM-optimized) NTRU Prime software vs. "3,520 bytes of RAM" used by (RAM-optimized) Kyber software. This refers to RAM used *during a computation* and reused for the next computation, not per-key RAM; this is outweighed by the difference between $5856N$ and $1518N$ as soon as $N \geq 3$.
- NTRU Prime, like Kyber, has the flexibility to precompute transforms of multiplication inputs, so if enough RAM is available for expanded keys then NTRU Prime will save time too.

In short, Kyber starts out using more RAM (assuming N is not very small), more communication, and more cycles than NTRU Prime. The argument that Kyber should be able to use fewer cycles than NTRU Prime relies on (1) using even more RAM and (2) ignoring the flexibility of NTRU Prime to also exploit extra RAM. Furthermore, even if Kyber is allowed to use much more RAM while NTRU Prime is not, Kyber uses so much more communication that it will still be slower than NTRU Prime.

7.10. More platforms. There are many more platforms of interest, and many variations in cost metrics of interest. The ecosystem of optimized NTRU Prime implementations is expanding to include more and more targets. Here are some examples.

FPGAs: When an application needs to fit many different processing units into the space available on an FPGA, the space consumed by each unit is critical. Speed matters only for units that are bottlenecks in the application. Marotzke [**251**] presented a *complete* constant-time `sntrup761` implementation that fits into a small corner of an Artix-7 FPGA (the FPGA designated by NIST for comparisons) with better throughput than needed for any known application of FPGAs. In case applications do need better speeds, very recent results in [**312**] show that NTRU Prime supports a wide range of area-throughput tradeoffs on the same Artix-7 FPGA:

- keygen fits into 629367 cycles in 7579 LUTs or 64026 cycles in 39200 LUTs,
- enc fits into 29245 cycles in 6379 LUTs or 5007 cycles in 40879 LUTs, and
- dec fits into 85628 cycles in 6279 LUTs or 10989 cycles in 36789 LUTs,

in all cases running at $\geq$131MHz. The combined keygen-enc-dec units in [**312**] are not much larger than single units. On a Zynq Ultrascale+, LUT counts are smaller and frequencies are higher, competitive with recent `ntruhps` Ultrascale+ results [**129**] for that FPGA. It is interesting to note that the current FPGA speed records for `ntruhps`, `sntrup`, and `saber` do not use NTTs, even though these FPGAs include multipliers.

ASICs: Central predictors of hardware efficiency, such as energy consumption, include bit operations—weighted by the cost of each bit operation in hardware—and communication costs. Investigations are underway regarding the number of bit operations required for all sizes of `sntrup` and `ntrulpr`. See Section 6.5.

Smartphones: There are many different smartphone CPUs. The main targets are 32-bit ARMv7-A and 64-bit ARMv8-A. Each of these targets includes many variations in microarchitectures. Work is in progress to optimize NTRU Prime software for these targets: for example, `sntrup761` decapsulation currently runs in 212981 Cortex-A72 cycles.

Software memory usage: Given how little space NTRU Prime uses on an FPGA, it is safe to extrapolate that NTRU Prime can also be squeezed into very little space in software, including code size, stack usage, etc. Work is underway to construct such software, both for the single-parameter-set scenario and for the multiple-parameter-set scenario. This may be useful for some applications on small devices.

Side-channel protection: All supported NTRU Prime implementations are protected against timing attacks, but many applications also need protection against power attacks, electromagnetic attacks, etc. A twist in the story here is 2021 Ngo–Dubrova–Guo–Johansson [**280**], an attack very efficiently breaking the *masked* implementation of SABER from [**38**]: the attack recovers the user's secret key from just 16 decapsulation traces. This attack, together with other recent side-channel attacks, has called into question the entire idea that these KEMs can be protected by low-order masking; see generally [**28**]. Investigations

Haswell cycles/key	enc	dec	`sntrup761` report
>6000000	59456	97684	round-1 submission
946772	55252	70464	round-2 update talk
156317	46914	56241	round-3 update talk

Haswell cycles/key	enc	dec	`ntrulpr761` report
47396	77280	95316	round-2 update talk
42515	69103	82071	round-3 update talk

Table 7.12. History of improvements in Haswell cycles for keygen, enc, and dec for the dimension recommended in the NTRU Prime submission. See [66].

are underway regarding automated conversions of sequences of bit operations for NTRU Prime into *high-order* masked implementations.

7.11. Key generation, part 1: speedups. It's far from clear that keygen time matters in these systems (see Section 6.3), so optimizing keygen hasn't been the top priority—but there has nevertheless been some work. This work has dramatically sped up `sntrup` keygen, so any concerns that might have been raised at the beginning regarding keygen time are now obsolete *even if* the application is bottlenecked by keygen time.

To see the speedups, let's focus specifically on

- Intel Haswell (NIST's designated large CPU for comparison) and
- dimension 761 (the dimension that has always been recommended in the NTRU Prime submission)

since this is where the longest history of performance data is available.

The round-1 submission reported key generation taking more than 6 million cycles; see Table 7.12. The round-2 update talk reported key generation more than 6 times faster. The round-3 update talk reported key generation another 6 times faster. See [66], including a web-browsing demo on top of TLS 1.3 using `sntrup761` with fast key generation.

Table 7.12 also covers `ntrulpr761`, the Product NTRU option in the NTRU Prime submission. This has faster keygen than `sntrup761` and smaller key sizes, but slower enc, slower dec, and larger ciphertexts. The round-3 cycle counts in the table allow a direct speed comparison between Product NTRU and Quotient NTRU: the ring is the same, there is extensive sharing of optimized software between the options, and the subroutines specific to each option have also been rewritten for performance.

The numbers show that, as soon as a key is used for 3 ciphertexts, `sntrup761` uses fewer cycles than `ntrulpr761`. It also sends fewer bytes, since its ciphertexts are smaller. A popular server broadcasting a key for clients around the Internet to use for the next few minutes is going to have the key used for many more than 3 ciphertexts.

7.13. Key generation, part 2: comparison to Kyber. Let's take Kyber's favorite scenario: N keys are used for only N ciphertexts; the sender and receiver each have an Intel CPU; also, let's ignore `ntrulpr`. Here are the full costs of two options, accounting for the latest benchmarks for publicly available software.

For `kyber768`, the receiver spends

- $1184N$ bytes sending keys plus $1088N$ bytes receiving ciphertexts, and
- $44178N$ cycles on key generation plus $47766N$ cycles on decapsulation;

total cost $2360944N$ cycles in the model of Section 6.6. The sender spends

- $1184N$ bytes receiving keys plus $1088N$ bytes sending ciphertexts, and
- $60258N$ cycles on encapsulation;

total cost $2332258N$ cycles. The sender and receiver together spend $4693202N$ cycles.

For `sntrup653`, the receiver spends

- $994N$ bytes sending keys plus $897N$ bytes receiving ciphertexts, and
- $164260N$ cycles on key generation plus $55778N$ bytes on decapsulation;

total cost $2111038N$ cycles. The sender spends

- $994N$ bytes receiving keys plus $897N$ bytes sending ciphertexts, and
- $44155N$ cycles on encapsulation;

total cost $1935155N$ cycles. The sender and receiver together spend $4046193N$ cycles. Notice that keygen is only 4% of this cost.

If the sender and receiver cannot afford the 16% extra cost of `kyber768` then for Kyber they have to jump down to `kyber512`, which has[23] Core-SVP only 2^{118}, exacerbating risks compared to `sntrup653`'s Core-SVP 2^{129}.

By taking a different cost limit one can construct an argument in the opposite direction, where the limit forces `sntrup` to a lower Core-SVP level than `kyber`. Keygen has negligible impact on these total-cost comparisons: the comparisons boil down to the sizes favoring `sntrup` at some security levels and favoring `kyber` at others, as in Figure 7.3.

7.14. Key generation, part 3: TLS. Typical pro-Kyber speed narratives select the new-key-for-every-ciphertext scenario as above, select an Intel CPU as above, implicitly select target security levels favorable to Kyber, and *ignore communication costs*. What is benchmarked is thus keygen+enc+dec on, e.g., Intel Haswell.

Kyber outperforms all the other small lattice KEMs in this benchmark, and advertises this as TLS performance. However, there is also ample evidence that this Kyber speedup doesn't matter, because *all* of the small lattice KEMs are much faster than necessary for this scenario. In other words, this scenario should be given very low weight in comparing the performance of these KEMs.

Let's look at the numbers. In 2017, Cloudflare [**214**] reported that 73% of its connections were TLS connections, and that 68% of its TLS key exchanges

[23] This once again gives `kyber512` the benefit of switching from Core-SVP to revised-Core-SVP.

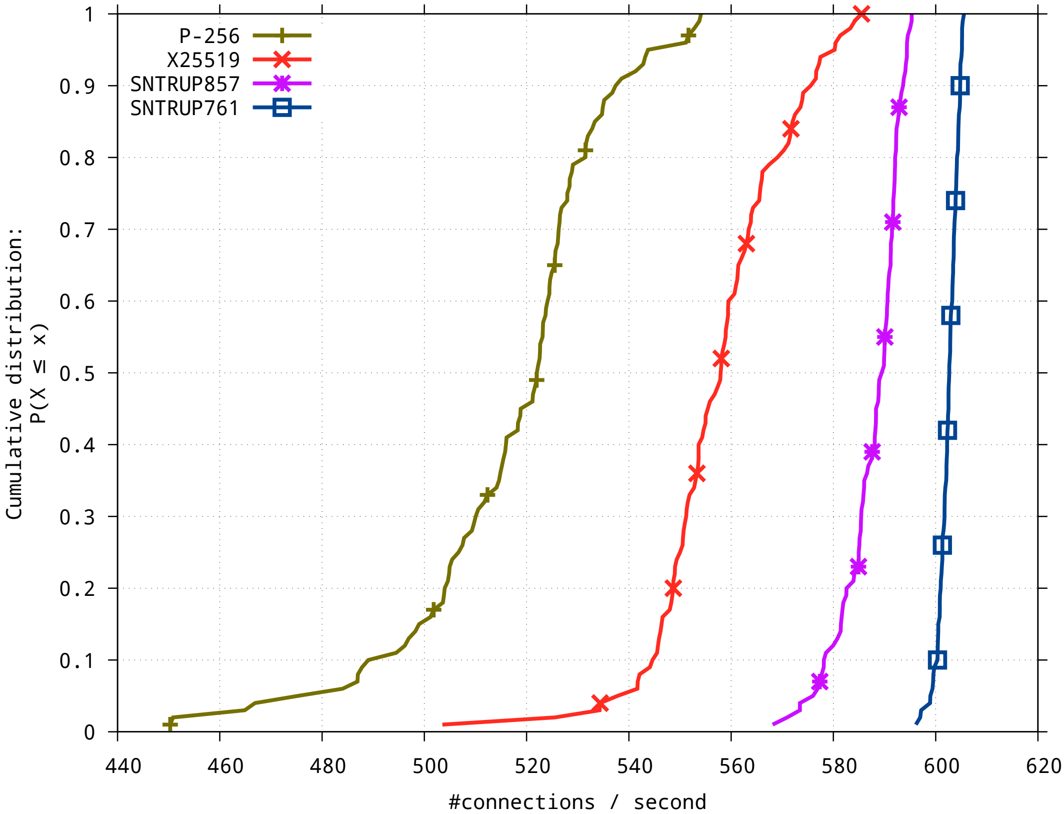

Fig. 7.15. Copy of [**66**, Figure 5]. Each curve in the figure shows, for one cryptosystem, the cumulative distribution of TLS 1.3 handshake speed observed in 100 experiments between two consumer-grade workstations over a fast local network. Each experiment measures the elapsed wall-clock time on the client side for 8192 sequentially established TLS 1.3 connections. The P-256 and X25519 experiments use the most optimized implementations available in OpenSSL 1.1.1 for pre-quantum ECC key exchange. The `sntrup761` and `sntrup857` experiments use the software from [**66**]. CPUs are Intel Core i7-6700 (client) and AMD Ryzen 7 2700X (server).

used the NIST P-256 elliptic curve. OpenSSL 1.1.1 (which is newer than [**214**]) takes 48000 Haswell cycles for NIST P-256 key generation and 235000 Haswell cycles for NIST P-256 scalar multiplication, in total 283000 cycles for Alice, and similarly 283000 cycles for Bob.

For comparison, the `sntrup761` Haswell software in [**66**] takes 259472 cycles for keygen+enc+dec. This is the *total* of Alice's time (212558 cycles) and Bob's time (46914 cycles), so `sntrup761` is more than twice as fast as NIST P-256. Beware that this comparison ignores the much higher communication costs of lattice systems compared to ECC, as noted in [**66**].

Macrobenchmarks of TLS connections easily see this speedup if one uses a fast local network, masking the communication costs. Figure 7.15, copied from [**66**], shows the results of an end-to-end experiment examining fully established TLS connections, including a new key for each connection on the client side, enc on the server side, dec on the client side, validation against a traditional RSA certificate, and communication over a local network. In this experiment, NIST P-256 completed 520 TLS connections/second, and `sntrup761` completed 600 TLS connections/second.

Does this speedup matter for the application? Cloudflare reported in [**214**] that only 1.8% of its CPU cycles were spent on TLS and concluded that "Using TLS is very cheap, even at the scale of Cloudflare". Most of the public-key cycles were spent on RSA. NIST P-256, handling not just 68% of the key exchanges but also 75% of the signatures, used just 8% of the cryptographic time, i.e., 0.15% of the total CPU cycles. Evidently handling *every* connection with TLS using NIST P-256 would have consumed only 0.22% of the total CPU cycles. A hybrid deployment, protecting each connection with `sntrup761` *and* NIST P-256, would still have consumed a negligible percentage of the total CPU cycles.

Does it matter that, in the same Haswell keygen+enc+dec benchmark, `saber` uses 286511 cycles, `ntruhrss701` uses 359076 cycles (here one should expect speedups using the techniques of [**66**]), and `kyber768` uses 152202 cycles? No. Again, all of these small lattice KEMs are much faster than necessary for this scenario. Increasing security levels would increase the KEM costs, but runs into communication bottlenecks long before running into computation bottlenecks; see Section 6.6.

Today most web browsing uses X25519, which is faster than NIST P-256. Does this mean that NIST P-256 is a bottleneck on Intel CPUs? No. The move to X25519 has been driven much more by other factors, such as the speed of X25519 in constrained environments (Apple deployed X25519 for ARM devices starting in 2010), the advanced X25519 software ecosystem (see, e.g., [**332**]), and security concerns regarding NSA-generated elliptic curves (see, e.g., [**68**]).

7.16. Performance overview and outlook. Ultimately what matters is whether users around the world can upgrade to post-quantum cryptography without having to compromise security. Google's first post-quantum experiment in 2016 concluded [**224**] that `newhope1024` would be "practical to quickly deploy". The Google–Cloudflare TLS experiment [**217**] showed that the communication costs of `ntruhrss701` had little impact on TLS, and the computation costs had even less impact, as one would expect from Sections 6.6 and 7.14.

Compared to `newhope` and `ntruhrss`, one obtains better tradeoffs between size and Core-SVP from NTRU's `ntruhps`, NTRU Prime's `sntrup`, SABER, and Kyber. See [**50**, Figure 3.5] and Figure 7.3. The highest Core-SVP in Figure 7.3 could be from `sntrup`, from `ntruhps`, or from `saber` (sometimes matched by `kyber`), depending on the application's exact size limit.

These TLS experiments did not include post-quantum authentication. Using post-quantum KEMs for authentication, as in [**45**], [**48**], and [**333**], is more efficient than using post-quantum signatures, and can be expected to double or triple the KEM costs per connection, depending on the number of authentication layers. On the other hand, servers distributing their long-term keys and per-minute keys through a broadcast network such as DNS will eliminate almost all of the costs of key distribution. These improvements should put much lower weight on public-key size than on ciphertext size, and much lower weight on key-generation time than on encapsulation time and decapsulation time. In any case, communication costs will remain dominant.

There are more reasons to think that cycle counts are an issue for applications running on small devices. This is most likely to be a problem for Kyber, since the way Kyber integrates NTTs for speedups on Intel CPUs creates cross-platform slowdowns. Applications building post-quantum coprocessors to address speed bottlenecks are likely to obtain the best efficiency from `ntruhps` and `sntrup`, somewhat worse efficiency from `saber`, and the worst efficiency from `kyber`. See Section 6.5.

A critical caveat for all of these KEMs is that the attack picture is unstable. See Section 1. The Core-SVP figures in the performance graphs consider only *known* attacks. Applications can and should react in two ways. The first is to take the largest dimensions they can afford; this gives a further advantage to NTRU and NTRU Prime, which scale better than SABER and Kyber. The second is to select a KEM family designed in light of the risks from *unknown* attacks. NTRU Prime's performance profile demonstrates that eliminating as many of these risks as possible, subject to the requirement of being a small lattice KEM, is compatible with attractive performance.

References

[1] — (no editor), *23rd annual network and distributed system security symposium, NDSS 2016, San Diego, California, USA, February 21–24, 2016*, The Internet Society, 2016. URL: `https://www.ndss-symposium.org/ndss2016/`. See [74].

[2] Amin Abdulrahman, Jiun-Peng Chen, Yu-Jia Chen, Vincent Hwang, Matthias J. Kannwischer, Bo-Yin Yang, *Multi-moduli NTTs for Saber on Cortex-M3 and Cortex-M4*, to appear, IACR Transactions on Cryptographic Hardware and Embedded Systems (2021). URL: `https://eprint.iacr.org/2021/995`. Citations in this document: §7.8.

[3] Luca Aceto, Monika Henzinger, Jirí Sgall (editors), *Automata, languages and programming—38th international colloquium, ICALP 2011, Zurich, Switzerland, July 4–8, 2011, proceedings, part I*, Springer, 2011. ISBN 978-3-642-22005-0. See [22].

[4] Carlisle Adams, Jan Camenisch (editors), *Selected areas in cryptography—SAC 2017, 24th international conference, Ottawa, ON, Canada, August 16–18, 2017, revised selected papers*, Lecture Notes in Computer Science, 10719, Springer, 2018. ISBN 978-3-319-72564-2. See [69].

[5] Carlisle M. Adams, Ali Miri, Michael J. Wiener (editors), *Selected areas in cryptography, 14th international workshop, SAC 2007, Ottawa, Canada, August 16–17, 2007, revised selected papers*, Springer, 2007. ISBN 978-3-540-77359-7. URL: `https://doi.org/10.1007/978-3-540-77360-3`. See [295], [348].

[6] Mete Akgün, Pinar Kavak, Hüseyin Demirci, *New results on the key scheduling algorithm of RC4*, in Indocrypt 2008 [**115**] (2008), 40–52. URL: `https://doi.org/10.1007/978-3-540-89754-5_4`. Citations in this document: §1.3.

[7] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela

Robinson, Daniel Smith-Tone, *Status report on the first round of the NIST Post-Quantum Cryptography Standardization Process*, NISTIR 8240 (2019). URL: https://csrc.nist.gov/publications/detail/nistir/8240/final. Citations in this document: §3.13.

[8] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, *Status report on the second round of the NIST Post-Quantum Cryptography Standardization Process*, NISTIR 8309 (2020). URL: https://csrc.nist.gov/publications/detail/nistir/8309/final. Citations in this document: §2.5, §3.16, §5.9, §6.8, §6.8.

[9] Martin R. Albrecht, Shi Bai, Pierre-Alain Fouque, Paul Kirchner, Damien Stehlé, Weiqiang Wen, *Faster enumeration-based lattice reduction: Root Hermite factor $k^{1/2k}$ time $k^{k/8+o(k)}$*, in Crypto 2020 [**260**] (2020), 186–212. URL: https://eprint.iacr.org/2020/707. Citations in this document: §1, §1, §B.4.

[10] Martin R. Albrecht, Shi Bai, Jianwei Li, Joe Rowell, *Lattice reduction with approximate enumeration oracles—practical algorithms and concrete performance*, in Crypto 2021 [**247**] (2021), 732–759. URL: https://eprint.iacr.org/2020/1260. Citations in this document: §1, §1, §B.4, §B.4.

[11] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, Ludovic Perret, *Algebraic algorithms for LWE problems*, ACM Communications on Computer Algebra **49** (2015), 62. URL: https://eprint.iacr.org/2014/1018. Citations in this document: §3.14.

[12] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, Thomas Wunderer, *Estimate all the {LWE, NTRU} Schemes!*, in SCN 2018 [**98**] (2018), 351–367. URL: https://eprint.iacr.org/2018/331. Citations in this document: §3.4, §3.4, §3.4, §7.2.

[13] Martin R. Albrecht, Christian Hanser, Andrea Höller, Thomas Pöppelmann, Fernando Virdia, Andreas Wallner, *Implementing RLWE-based schemes using an RSA co-processor*, IACR Transactions on Cryptographic Hardware and Embedded Systems **2019.1** (2019), 169–208. URL: https://eprint.iacr.org/2018/425. Citations in this document: §6.5.

[14] Nadhem J. AlFardan, Daniel J. Bernstein, Kenneth G. Paterson, Bertram Poettering, Jacob C. N. Schuldt, *On the security of RC4 in TLS*, in USENIX Security 2013 [**205**] (2013), 305–320. URL: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/alFardan. Citations in this document: §1.3.

[15] Erdem Alkim, Joppe W. Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Chris Peikert, Ananth Raghunathan, Douglas Stebila, *FrodoKEM: Learning with errors key encapsulation* (2017). URL: https://web.archive.org/web/20211025061246/https://frodokem.org/files/FrodoKEM-specification-20171130.pdf. Citations in this document: §1, §3.10, §A.3.

[16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, Peter Schwabe, *Post-quantum key exchange—a new hope*, in USENIX Security 2016 [**179**] (2016), 327–343. URL: https://eprint.iacr.org/2015/1092. Citations in this document: §3.5, §A.1.

[17] Andris Ambainis, *Quantum walk algorithm for element distinctness*, SIAM Journal on Computing **37** (2007), 210–239. URL: https://arxiv.org/abs/quant-ph/0311001. Citations in this document: §B.2.

[18] Yoshinori Aono, Phong Q. Nguyen, Yixin Shen, *Quantum lattice enumeration and tweaking discrete pruning*, in Asiacrypt 2018 [**313**] (2018), 405–434. URL: https://eprint.iacr.org/2018/546. Citations in this document: §1, §1, §1, §B.4.

[19] Daniel Apon, *RE: On Recommended Hardware*, email dated 6 Feb 2019 19:10:10 +0000 (2019). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/cJxMq0_90gU/m/qbGEs3TXGwAJ. Citations in this document: §6.2, §7.4.

[20] Daniel Apon, *Passing the final checkpoint! NIST PQC 3rd round begins* (2020). URL: https://archive.is/RG14h. Citations in this document: §2.5, §4.5, §4.5, §5.3, §5.3, §5.3.

[21] Daniel Apon, *Re: S-unit attacks*, email dated 24 Aug 2021 07:45:56 -0700 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/3mVeyEfYnUY/m/_xYg6HRcDQAJ. Citations in this document: §5.9.

[22] Sanjeev Arora, Rong Ge, *New algorithms for learning in presence of errors*, in ICALP 2011 [**3**] (2011), 403–415. URL: https://web.archive.org/web/20210829030507/https://users.cs.duke.edu/~rongge/LPSN.pdf. Citations in this document: §3.14.

[23] Derek Atkins, *Re: Kyber's inefficiency: some data points*, email dated 20 Oct 2021 18:53:00 +0000 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/ik1pXoAZk8s/m/exzVjOHrAAAJ. Citations in this document: §7.8, §7.8.

[24] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé, *CRYSTALS-Kyber: Algorithm specifications and supporting documentation* (2017). URL: https://web.archive.org/web/20210807154855/https://pq-crystals.org/kyber/data/kyber-specification.pdf. Citations in this document: §1.3, §1.3, §B.3, §B.4.

[25] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé, *CRYSTALS-Kyber: Algorithm specifications and supporting documentation* (2020). URL: https://web.archive.org/web/20211007045636/https://pq-crystals.org/kyber/data/kyber-specification-round3.pdf. Citations in this document: §1.3, §1.3, §3.4, §3.4, §3.4, §5.7, §6.4, §A.3, §A.3.

[26] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé, *CRYSTALS–Kyber* (2021). URL: https://www.nist.gov/video/third-pqc-standardization-conference-session-vii-performance-and-candidate-updates. Citations in this document: §1.3, §1.3, §6.4, §6.5, §A.1.

[27] Roberto Avanzi, Howard M. Heys (editors), *Selected areas in cryptography—SAC 2016—23rd international conference, St. John's, NL, Canada, August 10–12, 2016, revised selected papers*, Springer, 2017. ISBN 978-3-319-69452-8. See [105], [340].

[28] Melissa Azouaoui, Joppe Bos, Olivier Bronchain, Christine Cloostermans, Björn Fay, Clement Hoffmann, Olivier Pereira, Thomas Peters, Joost Renes, Tobias Schneider, François-Xavier Standaert, *Side-channel-assisted key recovery attacks on IND-CCA transformations*, email dated 1 Sep 2021 19:44:13 +0000 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/iVbJkCytoog/m/2xnezD2IDgAJ. Citations in this document: §7.10, §A.1.

[29] Steve Babbage, Dario Catalano, Louis Granboulan, Arjen Lenstra, Christof Paar, Jan Pelzl, Thomas Pornin, Bart Preneel, Matt Robshaw, Andy Rupp,

Nigel Smart, Michael Ward, *ECRYPT yearly report on algorithms and keysizes* (2005). URL: https://web.archive.org/web/20170705114342/https://www.ecrypt.eu.org/ecrypt1/documents/D.SPA.10-1.1.pdf. Citations in this document: §2.5.

[30] Aditya Bagchi, Indrakshi Ray (editors), *Information systems security—9th international conference, ICISS 2013, Kolkata, India, December 16–20, 2013, proceedings*, Springer, 2013. ISBN 978-3-642-45203-1. URL: https://doi.org/10.1007/978-3-642-45204-8. See [242].

[31] Shi Bai, Damien Stehlé, Weiqiang Wen, *Measuring, simulating and exploiting the head concavity phenomenon in BKZ*, in Asiacrypt 2018 [**313**] (2018), 369–404. URL: https://eprint.iacr.org/2018/856. Citations in this document: §1.

[32] Abhishek Banerjee, Chris Peikert, Alon Rosen, *Pseudorandom functions and lattices*, in Eurocrypt 2012 [**315**] (2012), 719–737. URL: https://eprint.iacr.org/2011/401. Citations in this document: §5.3, §5.3, §5.3.

[33] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, Emmanuel Thomé, *A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic*, in Eurocrypt 2014 [**281**] (2014), 1–16. URL: https://eprint.iacr.org/2013/400. Citations in this document: §2.3.

[34] Andrea Basso, Jose Maria Bermudo Mera, Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Michiel Van Beirendonck, Frederik Vercauteren, *SABER: Mod-LWR based KEM (round 3 submission)* (2020). URL: https://www.esat.kuleuven.be/cosic/pqcrypto/saber/files/saberspecround3.pdf. Citations in this document: §A.1, §A.1.

[35] Riddhipratim Basu, Shirshendu Ganguly, Subhamoy Maitra, Goutam Paul, *A complete characterization of the evolution of RC4 pseudo random generation algorithm*, Journal of Mathematical Cryptology **2** (2008), 257–289. URL: https://doi.org/10.1515/JMC.2008.012. Citations in this document: §1.3.

[36] Riddhipratim Basu, Subhamoy Maitra, Goutam Paul, Tanmoy Talukdar, *On some sequences of the secret pseudo-random index j in RC4 key scheduling*, in AAECC 2009 [**89**] (2009), 137–148. URL: https://doi.org/10.1007/978-3-642-02181-7_15. Citations in this document: §1.3.

[37] Jens Bauch, Daniel J. Bernstein, Henry de Valence, Tanja Lange, Christine van Vredendaal, *Short generators without quantum computers: the case of multiquadratics*, in Eurocrypt 2017 [**121**] (2017), 27–59. URL: https://eprint.iacr.org/2017/404. Citations in this document: §2.5.

[38] Michiel Van Beirendonck, Jan-Pieter D'Anvers, Angshuman Karmakar, Josep Balasch, Ingrid Verbauwhede, *A side-channel-resistant implementation of SABER*, ACM Journal on Emerging Technologies in Computing Systems **17** (2021), 10:1–10:26. URL: https://eprint.iacr.org/2020/733. Citations in this document: §7.10.

[39] Mihir Bellare, Hannah Davis, Felix Günther, *Separate your domains: NIST PQC KEMs, oracle cloning and read-only indifferentiability*, in Eurocrypt 2020 [**97**] (2020), 3–32. URL: https://eprint.iacr.org/2020/241. Citations in this document: §1, §A.1.

[40] Olivier Bernard, Adeline Roux-Langlois, *Twisted-PHS: using the product formula to solve approx-SVP in ideal lattices*, in Asiacrypt 2020 [**273**] (2020), 349–380. URL: https://eprint.iacr.org/2020/1081. Citations in this document: §1.

[41] Daniel J. Bernstein, *Multidigit multiplication for mathematicians* (2001). URL: https://cr.yp.to/papers.html#m3. Citations in this document: §6.5.

[42] Daniel J. Bernstein, *Curve25519: new Diffie-Hellman speed records*, in PKC 2006 [**361**] (2006), 207–228. URL: https://cr.yp.to/papers.html#curve25519. Citations in this document: §2.5, §6.1, §6.1.

[43] Daniel J. Bernstein, *Complexity news: discrete logarithms in multiplicative groups of small-characteristic finite fields—the algorithm of Barbulescu, Gaudry, Joux, Thomé* (2013). URL: https://cr.yp.to/talks/2013.07.18/slides-djb-20130718-a4.pdf. Citations in this document: §2.5.

[44] Daniel J. Bernstein, *A subfield-logarithm attack against ideal lattices* (2014). URL: https://blog.cr.yp.to/20140213-ideal.html. Citations in this document: §2.2, §2.5, §2.5, §5.5.

[45] Daniel J. Bernstein, *The post-quantum Internet* (2016). URL: https://cr.yp.to/talks.html#2016.02.24. Citations in this document: §7.16.

[46] Daniel J. Bernstein, *Some challenges in post-quantum standardization* (2016). URL: https://blog.cr.yp.to/20161030-pqnist.html. Citations in this document: §A.1, §B.1, §B.1, §B.5.

[47] Daniel J. Bernstein, *Re: NIST request for feedback on measurement of security strengths in our upcoming PQC call for proposals*, email dated 20 Nov 2016 05:14:07 +0000 (2016). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/lBc-Gj2rSx0/m/vdBspd-uCgAJ. Citations in this document: §B.5.

[48] Daniel J. Bernstein, *Internet: Integration* (2018). URL: https://pqcrypto.eu.org/deliverables/d2.5.pdf. Citations in this document: §6.3, §7.16.

[49] Daniel J. Bernstein, *Comparing proofs of security for lattice-based encryption*, Second PQC Standardization Conference (2019). URL: https://cr.yp.to/papers.html#latticeproofs. Citations in this document: §3.2, §3.3, §3.9, §5.2, §5.7, §5.8, §5.9, §5.9, §A.1, §A.3.

[50] Daniel J. Bernstein, *Visualizing size-security tradeoffs for lattice-based encryption*, Second PQC Standardization Conference (2019). URL: https://cr.yp.to/papers.html#paretoviz. Citations in this document: §3.1, §6.6, §7.2, §7.2, §7.2, §7.16, §B.

[51] Daniel J. Bernstein, *Re: Why standardize "big" systems?*, email dated 5 Oct 2019 13:12:16 -0000 (2019). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/msRrR13muS4/m/ZBPrXvJvCwAJ. Citations in this document: §3.1.

[52] Daniel J. Bernstein, *Re: Low-Order Factors for NTRUPrime*, email dated 22 Nov 2019 08:46:00 -0000 (2019). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/2qABXjHl3mg/m/PLo6noJqAAAJ. Citations in this document: §5.6, §5.6.

[53] Daniel J. Bernstein, *A discretization attack* (2020). URL: https://cr.yp.to/papers.html#categories. Citations in this document: §1.3, §6.8, §7.2, §B.2, §B.4, §B.5.

[54] Daniel J. Bernstein, *ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER*, email dated 30 May 2020 02:15:31 +0200 (2020). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/o2roJXAlsUk/m/EHW9h87kAAAJ. Citations in this document: §1.3.

[55] Daniel J. Bernstein, *Hybrid attacks*, email dated 7 Jul 2020 19:06:42 +0200 (2020). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/8DvX3yF5bZc/m/CPJjiLbtAQAJ. Citations in this document: §3.5.

[56] Daniel J. Bernstein, *Re: ROUND 2 OFFICIAL COMMENT: NewHope*, email dated 25 Jul 2020 10:36:57 +0200 (2020). URL: https://groups.google.com/

a/list.nist.gov/g/pqc-forum/c/8_uKOBN4Srw/m/wiGLuW-tDAAJ. Citations in this document: §5.4.

[57] Daniel J. Bernstein, *Re: ROUND 3 OFFICIAL COMMENT: CRYSTALS-KYBER*, email dated 17 Dec 2020 19:30:46 +0100 (2020). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/NSe0wAzKJtA/m/ZHp_KJQ1AwAJ. Citations in this document: §1.3.

[58] Daniel J. Bernstein, *Patent-buyout updates*, email dated 17 May 2021 12:09:27 +0200 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/nbIZhtICKWU/m/ML7aYY71AgAJ. Citations in this document: §3.17, §3.17.

[59] Daniel J. Bernstein, *On the looseness of FO derandomization* (2021). URL: https://cr.yp.to/papers.html#footloose. Citations in this document: §3.8, §5.8, §5.8, §7.7.

[60] Daniel J. Bernstein, *S-unit attacks* (2021). URL: https://cr.yp.to/talks.html#2021.08.20. Citations in this document: §1.2, §1.2, §5.9.

[61] Daniel J. Bernstein, *Fast verified post-quantum software* (2021). URL: https://cr.yp.to/talks.html#2021.09.03. Citations in this document: §3.6, §A.3.

[62] Daniel J. Bernstein, *Re: ROUND 3 OFFICIAL COMMENT: NTRU Prime*, email dated 8 Oct 2021 14:17:37 +0200 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/zWxLiIAFG8c/m/zbr8OkTwBgAJ. Citations in this document: §3.17, §3.17.

[63] Daniel J. Bernstein, *Kyber's inefficiency: some data points*, email dated 20 Oct 2021 19:58:29 +0200 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/ik1pXoAZk8s/m/sKfmfrXMAQAJ. Citations in this document: §7.8, §7.8, §7.8.

[64] Daniel J. Bernstein, Leon Groot Bruinderink, Tanja Lange, Lorenz Panny, *HILA5 Pindakaas: On the CCA security of lattice-based encryption with error correction*, in Africacrypt 2018 [**198**] (2018), 203–216. URL: https://eprint.iacr.org/2017/1214. Citations in this document: §A.1.

[65] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, Bo-Yin Yang, *NTRU Prime: round 3* (2020). URL: https://ntruprime.cr.yp.to/nist.html. Citations in this document: §1.3, §3.4, §3.13, §6.3, §6.6, §6.6, §6.6, §7.7, §7.7, §A.1, §A.1, §A.1, §A.1, §A.3, §B, §B.3, §B.4.

[66] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Nicola Tuveri, *OpenSSLNTRU: Faster post-quantum TLS key exchange*, USENIX Security 2022, to appear (2021). URL: https://eprint.iacr.org/2021/826. Citations in this document: §6.3, §6.4, §7.12, §7.12, §7.11, §7.14, §7.14, §7.15, §7.15, §7.15, §7.15, §7.14, §7.14, §A.1, §A.1, §A.1, §A.3.

[67] Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen (editors), *Post-quantum cryptography*, Springer, 2009. ISBN 978-3-540-88701-0. See [259].

[68] Daniel J. Bernstein, Tung Chou, Chitchanok Chuengsatiansup, Andreas Hülsing, Eran Lambooij, Tanja Lange, Ruben Niederhagen, Christine van Vredendaal, *How to manipulate curve standards: a white paper for the black hat*, in SSR 2015 [**113**] (2015), 109–139. URL: https://bada55.cr.yp.to. Citations in this document: §7.14.

[69] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, Christine van Vredendaal, *NTRU Prime: reducing attack surface at low cost*, in SAC 2017 [**4**] (2017), 235–260. URL: https://ntruprime.cr.yp.to/papers.html. Citations in this document: §3.2, §3.12, §A.1, §A.1, §A.1.

[70] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, Tanja Lange, *Elligator: elliptic-curve points indistinguishable from uniform random strings*, in CCS 2013 [**324**] (2013), 967–980. URL: https://eprint.iacr.org/2013/325. Citations in this document: §A.1.

[71] Daniel J. Bernstein, Tanja Lange (editors), *eBACS: ECRYPT Benchmarking of Cryptographic Systems*, accessed 15 October 2021 (2021). URL: https://bench.cr.yp.to. Citations in this document: §6.4.

[72] Daniel J. Bernstein, Tanja Lange, *Non-randomness of S-unit lattices* (2021). URL: https://cr.yp.to/papers.html#spherical. Citations in this document: §1.2, §2.6, §5.9.

[73] Daniel J. Bernstein, Edoardo Persichetti, *Towards KEM unification* (2018). URL: https://cr.yp.to/papers.html#tightkem. Citations in this document: §5.8, §A.1.

[74] Karthikeyan Bhargavan, Gaëtan Leurent, *Transcript collision attacks: breaking authentication in TLS, IKE and SSH*, in NDSS 2016 [**1**] (2016). URL: https://hal.inria.fr/hal-01244855/. Citations in this document: §A.1.

[75] Karthikeyan Bhargavan, Elisabeth Oswald, Manoj Prabhakaran (editors), *Progress in cryptology—INDOCRYPT 2020—21st international conference on cryptology in India, Bangalore, India, December 13–16, 2020, proceedings*, Springer, 2020. ISBN 978-3-030-65276-0. URL: https://doi.org/10.1007/978-3-030-65277-7. See [102].

[76] Lei Bi, Xianhui Lu, Junjie Luo, Kunpeng Wang, Zhenfei Zhang, *Hybrid dual attack on LWE with arbitrary secrets* (2021). URL: https://eprint.iacr.org/2021/152. Citations in this document: §1.

[77] Jean-François Biasse, Claus Fieker, Tommy Hofmann, Aurel Page, *Norm relations and computational problems in number fields* (2020). URL: https://arxiv.org/abs/2002.12332. Citations in this document: §2.5.

[78] Jean-François Biasse, Fang Song, *Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields*, in SODA 2016 [**215**] (2016), 893–902. URL: https://epubs.siam.org/doi/10.1137/1.9781611974331.ch64. Citations in this document: §1.2.

[79] Eli Biham, Yaniv Carmeli, *Efficient reconstruction of RC4 keys from internal states*, in FSE 2008 [**285**] (2008), 270–288. URL: https://doi.org/10.1007/978-3-540-71039-4_17. Citations in this document: §1.3.

[80] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, Edoardo Persichetti, *Tighter proofs of CCA security in the quantum random oracle model*, in TCC 2019 [**178**] (2019), 61–90. URL: https://eprint.iacr.org/2019/590. Citations in this document: §5.7, §5.7.

[81] Alex Biryukov, Guang Gong, Douglas R. Stinson (editors), *Selected areas in cryptography—17th international workshop, SAC 2010, Waterloo, Ontario, Canada, August 12–13, 2010, revised selected papers*, Springer, 2011. ISBN 978-3-642-19573-0. URL: https://doi.org/10.1007/978-3-642-19574-7. See [334].

[82] Uri Blumenthal, *Re: Non-applicability of the CNRS patent to Kyber and Saber, and next steps*, email dated 14 Oct 2021 15:29:26 +0000 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/2Xv0mrF91Vo/m/pXC1BMneAQAJ. Citations in this document: §6.2.

[83] Carlo Blundo, Stelvio Cimato (editors), *Security in communication networks, 4th international conference, SCN 2004, Amalfi, Italy, September 8–10, 2004, revised selected papers*, Springer, 2005. ISBN 3-540-24301-1. See [338].

[84] Alexandra Boldyreva, Daniele Micciancio (editors), *Advances in cryptology—CRYPTO 2019—39th annual international cryptology conference, Santa Barbara, CA, USA, August 18–22, 2019, proceedings, part I*, Springer, 2019. ISBN 978-3-030-26947-0. See [138].

[85] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, Douglas Stebila, *Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE*, in CCS 2016 [**354**] (2016), 1006–1018. URL: https://eprint.iacr.org/2016/659. Citations in this document: §A.1.

[86] Colin Boyd, Juan Manuel González Nieto (editors), *Information security and privacy, 14th Australasian conference, ACISP 2009, Brisbane, Australia, July 1–3, 2009, proceedings*, Springer, 2009. ISBN 978-3-642-02619-5. URL: https://doi.org/10.1007/978-3-642-02620-1. See [266].

[87] Colin Boyd, Willy Susilo (editors), *8th Australasian information security conference 2010, AISC 2010, Brisbane, Australia, January 2010*, Australian Computer Society, 2010. ISBN 978-1-920682-86-6. URL: https://crpit.scem.westernsydney.edu.au/Vol105.html. See [358].

[88] Serdar Boztas, Hsiao-feng Lu (editors), *Applied algebra, algebraic algorithms and error-correcting codes, 17th international symposium, AAECC-17, Bangalore, India, December 16–20, 2007, proceedings*, Springer, 2007. ISBN 978-3-540-77223-1. URL: https://doi.org/10.1007/978-3-540-77224-8. See [297].

[89] Maria Bras-Amorós, Tom Høholdt (editors), *Applied algebra, algebraic algorithms and error-correcting codes, 18th international symposium, AAECC-18 2009, Tarragona, Catalonia, Spain, June 8–12, 2009, proceedings*, Springer, 2009. ISBN 978-3-642-02180-0. URL: https://doi.org/10.1007/978-3-642-02181-7. See [36].

[90] Gilles Brassard, Peter Høyer, Alain Tapp, *Quantum cryptanalysis of hash and claw-free functions*, in LATIN 1998 [**233**] (1998), 163–169. Citations in this document: §B.2.

[91] Richard P. Brent, H. T. Kung, *The area-time complexity of binary multiplication*, Journal of the ACM **28** (1981), 521–534. URL: https://wwwmaths.anu.edu.au/~brent/pub/pub055.html. Citations in this document: §B.

[92] Rémi Bricout, Sean Murphy, Kenneth G. Paterson, Thyla van der Merwe, *Analysing and exploiting the Mantin biases in RC4*, Designs, Codes and Cryptography **86** (2018), 743–770. URL: https://eprint.iacr.org/2016/063. Citations in this document: §1.3.

[93] Mike Brown, *Re: ROUND 2 OFFICIAL COMMENT: NewHope*, email dated 1 Jun 2019 19:59:45 +0000 (2019). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/P1B_InLVJmU/m/hWLNLNJRBgAJ. Citations in this document: §3.17.

[94] Joe P. Buhler, Hendrik W. Lenstra, Jr., Carl Pomerance, *Factoring integers with the number field sieve*, in [**228**] (1993), 50–94. URL: https://web.archive.org/web/20211027165847/https://pub.math.leidenuniv.nl/~lenstrahw/PUBLICATIONS/1993e/art.pdf. Citations in this document: §2.3.

[95] Jacques Calmet, Willi Geiselmann, Jörn Müller-Quade (editors), *Mathematical methods in computer science, MMICS 2008, Karlsruhe, Germany, December 17–19, 2008—essays in memory of Thomas Beth*, Springer, 2008. ISBN 978-3-540-89993-8. URL: https://doi.org/10.1007/978-3-540-89994-5. See [202].

[96] Peter Campbell, Michael Groves, Dan Shepherd, *Soliloquy: a cautionary tale* (2014). URL: https://web.archive.org/web/20211022132624/

https://docbox.etsi.org/Workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves_Annex.pdf. Citations in this document: §1.2, §2.5, §2.5, §2.5.

[97] Anne Canteaut, Yuval Ishai (editors), *Advances in cryptology—EUROCRYPT 2020—39th annual international conference on the theory and applications of cryptographic techniques, Zagreb, Croatia, May 10–14, 2020, proceedings, part II*, Springer, 2020. ISBN 978-3-030-45723-5. See [39].

[98] Dario Catalano, Roberto De Prisco (editors), *Security and cryptography for networks—11th international conference, SCN 2018, Amalfi, Italy, September 5–7, 2018, proceedings*, Lecture Notes in Computer Science, 11035, Springer, 2018. See [12].

[99] André Chailloux, Johanna Loyer, *Lattice sieving via quantum random walks* (2021). URL: https://eprint.iacr.org/2021/570. Citations in this document: §1, §1.

[100] Chandratop Chakraborty, Pranab Chakraborty, Subhamoy Maitra, *RC4: non-randomness in the index j and some results on its cycles*, in [**175**] (2019), 95–114. URL: https://doi.org/10.1007/978-3-030-35423-7_5. Citations in this document: §1.3.

[101] Chandratop Chakraborty, Pranab Chakraborty, Subhamoy Maitra, *Glimpses are forever in RC4 amidst the spectre of biases*, Discrete Applied Mathematics **298** (2021), 84–102. URL: https://eprint.iacr.org/2020/512. Citations in this document: §1.3.

[102] Pranab Chakraborty, Subhamoy Maitra, *More glimpses of the RC4 internal state array*, in Indocrypt 2020 [**75**] (2020), 294–311. URL: https://doi.org/10.1007/978-3-030-65277-7_13. Citations in this document: §1.3.

[103] Pranab Chakraborty, Subhamoy Maitra, *Further clarification on Mantin's digraph repetition bias in RC4*, Designs, Codes and Cryptography **89** (2021), 127–141. URL: https://eprint.iacr.org/2020/068. Citations in this document: §1.3.

[104] Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, Palash Sarkar, *Another look at tightness II: Practical issues in cryptography*, in Mycrypt 2016 [**314**] (2017), 21–55. URL: https://eprint.iacr.org/2016/360. Citations in this document: §5.9.

[105] Hao Chen, Kristin E. Lauter, Katherine E. Stange, *Security considerations for Galois non-dual RLWE families*, in SAC 2016 [**27**] (2016), 443–462. URL: https://eprint.iacr.org/2016/193. Citations in this document: §3.12.

[106] Hao Chen, Kristin E. Lauter, Katherine E. Stange, *Attacks on the search RLWE problem with small errors*, SIAM Journal on Applied Algebra and Geometry **1** (2017), 665–682. URL: https://eprint.iacr.org/2015/971. Citations in this document: §3.12, §3.12.

[107] Jiageng Chen, Atsuko Miyaji, *A new class of RC4 colliding key pairs with greater Hamming distance*, in ISPEC 2010 [**216**] (2010), 30–44. URL: https://doi.org/10.1007/978-3-642-12827-1_3. Citations in this document: §1.3.

[108] Jiageng Chen, Atsuko Miyaji, *Generalized RC4 key collisions and hash collisions*, in [**156**] (2010), 73–87. URL: https://doi.org/10.1007/978-3-642-15317-4_6. Citations in this document: §1.3.

[109] Jiageng Chen, Atsuko Miyaji, *A new practical key recovery attack on the stream cipher RC4 under related-key model*, in Inscrypt 2010 [**219**] (2010), 62–76. URL: https://doi.org/10.1007/978-3-642-21518-6_5. Citations in this document: §1.3.

[110] Jiageng Chen, Atsuko Miyaji, *Generalized analysis on key collisions of stream cipher RC4*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E94.A** (2011), 2194–2206. URL: https://doi.org/10.1587/transfun.E94.A.2194. Citations in this document: §1.3.

[111] Jiageng Chen, Atsuko Miyaji, *How to find short RC4 colliding key pairs*, in ISC 2011 [**220**] (2011), 32–46. URL: https://doi.org/10.1007/978-3-642-24861-0_3. Citations in this document: §1.3.

[112] Jiageng Chen, Atsuko Miyaji, *Novel strategies for searching RC4 key collisions*, Computers & Mathematics with Applications **66** (2013), 81–90. URL: https://doi.org/10.1016/j.camwa.2012.09.013. Citations in this document: §1.3.

[113] Liqun Chen, Shin'ichiro Matsuo (editors), *Security standardisation research—second international conference, SSR 2015, Tokyo, Japan, December 15–16, 2015, proceedings*, 9497, Springer, 2015. ISBN 978-3-319-27151-4. URL: https://doi.org/10.1007/978-3-319-27152-1. See [68].

[114] Jung Hee Cheon, Jean-Pierre Tillich (editors), *Post-quantum cryptography—12th international workshop, PQCrypto 2021, Daejeon, South Korea, July 20–22, 2021, proceedings*, Springer, 2021. ISBN 978-3-030-81292-8. See [180].

[115] Dipanwita Roy Chowdhury, Vincent Rijmen, Abhijit Das (editors), *Progress in cryptology—INDOCRYPT 2008, 9th international conference on cryptology in India, Kharagpur, India, December 14–17, 2008, proceedings*, Springer, 2008. ISBN 978-3-540-89753-8. URL: https://doi.org/10.1007/978-3-540-89754-5. See [6], [243].

[116] Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, Bo-Yin Yang, *NTT multiplication for NTT-unfriendly rings: new speed records for Saber and NTRU on Cortex-M4 and AVX2*, IACR Transactions on Cryptographic Hardware and Embedded Systems **2021.2** (2021), 159–188. URL: https://eprint.iacr.org/2020/1397. Citations in this document: §7.8.

[117] Carlos Cid, Christian Rechberger (editors), *Fast software encryption—21st international workshop, FSE 2014, London, UK, March 3–5, 2014, revised selected papers*, Springer, 2015. ISBN 978-3-662-46705-3. URL: https://doi.org/10.1007/978-3-662-46706-0. See [171].

[118] Henri Cohen, *A course in computational algebraic number theory*, Graduate Texts in Mathematics, 138, Springer, 1993. ISBN 3-540-55640-0. Citations in this document: §2.6, §2.6, §2.6, §2.6.

[119] Henri Cohen, *Advanced topics in computational number theory*, Graduate Texts in Mathematics, 193, Springer, 2000. ISBN 0-387-98727-4. Citations in this document: §2.6.

[120] Don Coppersmith, *Fast evaluation of logarithms in fields of characteristic two*, IEEE Transactions on Information Theory **30** (1984), 587–594. Citations in this document: §2.3, §2.3.

[121] Jean-Sébastien Coron, Jesper Buus Nielsen (editors), *Advances in Cryptology—EUROCRYPT 2017—36th annual international conference on the theory and applications of cryptographic techniques, Paris, France, April 30–May 4, 2017, proceedings, part I*, 2017. ISBN 978-3-319-56619-1. See [37], [124], [206].

[122] Ronald Cramer (editor), *Advances in cryptology—EUROCRYPT 2005, 24th annual international conference on the theory and applications of cryptographic techniques, Aarhus, Denmark, May 22–26, 2005, proceedings*, Springer, 2005. ISBN 3-540-25910-4. URL: https://doi.org/10.1007/b136415. See [249].

[123] Ronald Cramer, Léo Ducas, Chris Peikert, Oded Regev, *Recovering short generators of principal ideals in cyclotomic rings*, in Eurocrypt 2016 [**149**] (2016), 559–585. URL: https://eprint.iacr.org/2015/313. Citations in this document: §1.2.

[124] Ronald Cramer, Léo Ducas, Benjamin Wesolowski, *Short Stickelberger class relations and application to Ideal-SVP*, in Eurocrypt 2017 [**121**] (2017), 324–348. URL: https://eprint.iacr.org/2016/885. Citations in this document: §1.2.

[125] Jan-Pieter D'Anvers, Senne Batsleer, *Multitarget decryption failure attacks and their application to Saber and Kyber* (2021). URL: https://eprint.iacr.org/2021/193. Citations in this document: §1, §2.2.

[126] Jan-Pieter D'Anvers, Frederik Vercauteren, Ingrid Verbauwhede, *On the impact of decryption failures on the security of LWE/LWR based schemes* (2018). URL: https://eprint.iacr.org/2018/1089. Citations in this document: §1.

[127] Jan-Pieter D'Anvers, Frederik Vercauteren, Ingrid Verbauwhede, *The impact of error dependencies on Ring/Mod-LWE/LWR based schemes*, in PQCrypto 2019 [**135**] (2019), 103–115. URL: https://eprint.iacr.org/2018/1172. Citations in this document: §1.

[128] Dana Dachman-Soled, Léo Ducas, Huijing Gong, Mélissa Rossi, *LWE with side information: attacks and concrete security estimation*, in Crypto 2020 [**260**] (2020), 329–358. URL: https://eprint.iacr.org/2020/292. Citations in this document: §1.

[129] Viet Ba Dang, Kamyar Mohajerani, Kris Gaj, *High-Speed hardware architectures and fair FPGA benchmarking of CRYSTALS-Kyber, NTRU, and Saber* (2021). URL: https://web.archive.org/web/20211023071659/https://csrc.nist.gov/CSRC/media/Events/third-pqc-standardization-conference/documents/accepted-papers/gaj-high-speed-hardware-gmu-pqc2021.pdf. Citations in this document: §7.10.

[130] Apurba Das, Subhamoy Maitra, Goutam Paul, Santanu Sarkar, *Some combinatorial results towards state recovery attack on RC4*, in ICISS 2011 [**191**] (2011), 204–214. URL: https://doi.org/10.1007/978-3-642-25560-1_14. Citations in this document: §1.3.

[131] Ed Dawson, Andrew J. Clark, Colin Boyd (editors), *Information security and privacy, 5th Australasian conference, ACISP 2000, Brisbane, Australia, July 10–12, 2000, proceedings*, Springer, 2000. ISBN 3-540-67742-9. URL: https://doi.org/10.1007/10718964. See [164].

[132] Sabyasachi Dey, Santanu Sarkar, *Generalization of Roos bias in RC4 and some results on key-keystream relations*, Journal of Mathematical Cryptology **12** (2018), 43–56. URL: https://doi.org/10.1515/jmc-2016-0061. Citations in this document: §1.3.

[133] Sabyasachi Dey, Santanu Sarkar, *Settling the mystery of $Z_r = r$ in RC4*, Cryptography and Communications **11** (2019), 697–715. URL: https://eprint.iacr.org/2017/1072. Citations in this document: §1.3.

[134] Jintai Ding, *Solving LWE problem with bounded errors in polynomial time* (2010). URL: https://eprint.iacr.org/2010/558. Citations in this document: §3.14.

[135] Jintai Ding, Rainer Steinwandt (editors), *Post-quantum cryptography—10th international conference, PQCrypto 2019, Chongqing, China, May 8–10, 2019, revised selected papers*, Springer, 2019. ISBN 978-3-030-25509-1. See [127].

[136] Emmanouil Doulgerakis, Thijs Laarhoven, Benne de Weger, *Sieve, enumerate, slice, and lift: Hybrid lattice algorithms for SVP via CVPP*, in Africacrypt 2020 [**283**] (2020), 301–320. URL: https://eprint.iacr.org/2020/487. Citations in this document: §1.

[137] Léo Ducas, Alice Pellet-Mary, *Re: S-unit attacks*, email dated 24 Aug 2021 07:45:56 -0700 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/3mVeyEfYnUY/m/qgmqYdLVAQAJ. Citations in this document: §1.2, §1.2, §5.9.

[138] Léo Ducas, Maxime Plançon, Benjamin Wesolowski, *On the shortness of vectors to be found by the Ideal-SVP quantum algorithm*, in Crypto 2019 [**84**] (2019), 322–351. URL: https://eprint.iacr.org/2019/234. Citations in this document: §1.2, §1.2, §1.2, §1.2.

[139] Léo Ducas, Wessel van Woerden, *NTRU fatigue: how stretched is overstretched?*, Asiacrypt 2021, to appear (2021). URL: https://eprint.iacr.org/2021/999. Citations in this document: §3.13.

[140] Julien Duman, Kathrin Hövelmanns, Eike Kiltz, Vadim Lyubashevsky, Gregor Seiler, Dominique Unruh, *A thorough treatment of highly-efficient NTRU instantiations* (2021). URL: https://eprint.iacr.org/2021/1352. Citations in this document: §5.1, §6.2, §6.3, §6.3, §6.4, §6.4, §6.4, §6.4, §6.4.

[141] Orr Dunkelman (editor), *Fast software encryption, 16th international workshop, FSE 2009, Leuven, Belgium, February 22–25, 2009, revised selected papers*, Springer, 2009. ISBN 978-3-642-03316-2. URL: https://doi.org/10.1007/978-3-642-03317-9. See [254].

[142] Orr Dunkelman, Somitra Kumar Sanadhya (editors), *Progress in cryptology—INDOCRYPT 2016—17th international conference on cryptology in India, Kolkata, India, December 11–14, 2016, proceedings*, 2016. ISBN 978-3-319-49889-8. URL: https://doi.org/10.1007/978-3-319-49890-4. See [193].

[143] Aaron Dutle, Mariano M. Moscato, Laura Titolo, César A. Muñoz, Ivan Perez (editors), *NASA formal methods—13th international symposium, NFM 2021, virtual event, May 24–28, 2021, proceedings*, 12673, Springer, 2021. ISBN 978-3-030-76383-1. URL: https://doi.org/10.1007/978-3-030-76384-8. See [332].

[144] Kirsten Eisenträger, Sean Hallgren, Alexei Y. Kitaev, Fang Song, *A quantum algorithm for computing the unit group of an arbitrary degree number field*, in STOC 2014 [**336**] (2014), 293–302. URL: https://web.archive.org/web/20211010020701/https://www.personal.psu.edu/kxe8/unitgroup.pdf. Citations in this document: §2.6.

[145] Kirsten Eisenträger, Sean Hallgren, Kristin E. Lauter, *Weak instances of PLWE*, in SAC 2014 [**199**] (2014), 183–194. URL: https://eprint.iacr.org/2014/784. Citations in this document: §3.12.

[146] Yara Elias, Kristin E. Lauter, Ekin Ozman, Katherine E. Stange, *Provably weak instances of Ring-LWE*, in CRYPTO 2015 [**160**] (2015), 63–92. URL: https://eprint.iacr.org/2015/106. Citations in this document: §3.12.

[147] European Patent Office, *All documents: EP2537284*, accessed 25 October 2021 (2021). URL: https://register.epo.org/application?number=EP11712927&lng=en&tab=doclist. Citations in this document: §3.17.

[148] Wieland Fischer, Naofumi Homma (editors), *Cryptographic hardware and embedded systems—CHES 2017—19th international conference, Taipei, Taiwan, September 25–28, 2017, proceedings*, Lecture Notes in Computer Science, 10529, Springer, 2017. ISBN 978-3-319-66786-7. See [181].

[149] Marc Fischlin, Jean-Sébastien Coron (editors), *Advances in Cryptology—EUROCRYPT 2016—35th annual international conference on the theory and applications of cryptographic techniques, Vienna, Austria, May 8–12, 2016, proceedings, part I*, Springer, 2016. ISBN 978-3-662-49889-7. See [123].

[150] Scott R. Fluhrer, Itsik Mantin, Adi Shamir, *Weaknesses in the key scheduling algorithm of RC4*, in SAC 2001 [**349**] (2001), 1–24. URL: https://doi.org/10.1007/3-540-45537-X_1. Citations in this document: §1.3.

[151] Scott R. Fluhrer, David A. McGrew, *Statistical analysis of the alleged RC4 keystream generator*, in FSE 2000 [**331**] (2000), 19–30. URL: https://doi.org/10.1007/3-540-44706-7_2. Citations in this document: §1.3.

[152] Ernest Foo, Douglas Stebila (editors), *Information security and privacy—20th Australasian conference, ACISP 2015, Brisbane, QLD, Australia, June 29–July 1, 2015, proceedings*, Springer, 2015. ISBN 978-3-319-19961-0. URL: https://doi.org/10.1007/978-3-319-19962-7. See [186].

[153] Eiichiro Fujisaki, Tatsuaki Okamoto, *Secure integration of asymmetric and symmetric encryption schemes*, in Crypto 1999 [**356**] (1999), 537–554. Citations in this document: §3.15.

[154] Walter Fumy (editor), *Advances in cryptology—EUROCRYPT '97, international conference on the theory and application of cryptographic techniques, Konstanz, Germany, May 11–15, 1997, proceeding*, Springer, 1997. ISBN 3-540-62975-0. URL: https://doi.org/10.1007/3-540-69053-0. See [163].

[155] Steven D. Galbraith, Shiho Moriai (editors), *Advances in cryptology—ASIACRYPT 2019—25th international conference on the theory and application of cryptology and information security, Kobe, Japan, December 8–12, 2019, proceedings, part I*, Springer, 2019. ISBN 978-3-030-34577-8. See [170].

[156] Juan A. Garay, Roberto De Prisco (editors), *Security and cryptography for networks, 7th international conference, SCN 2010, Amalfi, Italy, September 13–15, 2010, proceedings*, Springer, 2010. ISBN 978-3-642-15316-7. URL: https://doi.org/10.1007/978-3-642-15317-4. See [108].

[157] Sanjam Garg, Craig Gentry, Shai Halevi, *Candidate multilinear maps from ideal lattices*, in Eurocrypt 2013 [**196**] (2013), 1–17. URL: https://eprint.iacr.org/2012/610. Citations in this document: §1.2.

[158] Christina Garman, Kenneth G. Paterson, Thyla van der Merwe, *Attacks only get better: Password recovery attacks against RC4 in TLS*, in USENIX Security 2015 [**200**] (2015), 113–128. URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/garman. Citations in this document: §1.3.

[159] Fletcher Gates, *Reduction-respecting parameters for lattice-based cryptosystems* (2018). URL: https://macsphere.mcmaster.ca/handle/11375/24466. Citations in this document: §5.9.

[160] Rosario Gennaro, Matthew Robshaw (editors), *Advances in cryptology—CRYPTO 2015—35th annual cryptology conference, Santa Barbara, CA, USA, August 16–20, 2015, proceedings, part I*, Springer, 2015. ISBN 978-3-662-47988-9. See [146].

[161] Craig Gentry, *Fully homomorphic encryption using ideal lattices*, in STOC 2009 [**265**] (2009), 169–178. URL: https://dl.acm.org/doi/abs/10.1145/1536414.1536440. Citations in this document: §1.2.

[162] Craig Gentry, Shai Halevi, *Implementing Gentry's fully-homomorphic encryption scheme*, in Eurocrypt 2011 [**292**] (2011), 129–148. URL: https://eprint.iacr.org/2010/520. Citations in this document: §1.2.

[163] Jovan Dj. Golic, *Linear statistical weakness of alleged RC4 keystream generator*, in [**154**] (1997), 226–238. URL: https://doi.org/10.1007/3-540-69053-0_16. Citations in this document: §1.3.

[164] Jovan Dj. Golic, *Iterative probabilistic cryptanalysis of RC4 keystream generator*, in ACISP 2000 [**131**] (2000), 220–233. URL: https://doi.org/10.1007/10718964_18. Citations in this document: §1.3.

[165] Jovan Dj. Golic, Guglielmo Morgari, *Iterative probabilistic reconstruction of RC4 internal states* (2008), 348. URL: https://eprint.iacr.org/2008/348. Citations in this document: §1.3.

[166] Daniel M. Gordon, *Discrete logarithms in $GF(p)$ using the number field sieve*, SIAM Journal on Discrete Mathematics **6** (1993), 124–138. Citations in this document: §2.3.

[167] Robert Granger, Antoine Joux, *Computing discrete logarithms* (2021). URL: https://eprint.iacr.org/2021/1140. Citations in this document: §2.3.

[168] Aurélien Greuet, *Smartcard and post-quantum crypto* (2021). URL: https://csrc.nist.gov/CSRC/media/Presentations/smartcard-and-post-quantum-crypto/images-media/session-5-greuet-smartcard-pqc.pdf. Citations in this document: §6.6, §7.8.

[169] Qian Guo, Thomas Johansson, Alexander Nilsson, *A key-recovery timing attack on post-quantum primitives using the Fujisaki-Okamoto transformation and its application on FrodoKEM*, in Crypto 2020 [**260**] (2020), 359–386. URL: https://eprint.iacr.org/2020/743. Citations in this document: §1, §3.6, §A.3.

[170] Qian Guo, Thomas Johansson, Jing Yang, *A novel CCA attack using decryption errors against LAC*, in Asiacrypt 2019 [**155**] (2019), 82–111. URL: https://eprint.iacr.org/2019/1308. Citations in this document: §1.

[171] Sourav Sen Gupta, Subhamoy Maitra, Willi Meier, Goutam Paul, Santanu Sarkar, *Dependence in IV-related bytes of RC4 key enhances vulnerabilities in WPA*, in FSE 2014 [**117**] (2014), 350–369. URL: https://doi.org/10.1007/978-3-662-46706-0_18. Citations in this document: §1.3.

[172] Sourav Sen Gupta, Subhamoy Maitra, Goutam Paul, Santanu Sarkar, *Proof of empirical RC4 biases and new key correlations*, in SAC 2011 [**262**] (2011), 151–168. URL: https://doi.org/10.1007/978-3-642-28496-0_9. Citations in this document: §1.3.

[173] Sourav Sen Gupta, Subhamoy Maitra, Goutam Paul, Santanu Sarkar, *(Non-)random sequences from (non-)random permutations—analysis of RC4 stream cipher* (2014), 67–108. URL: https://eprint.iacr.org/2011/448. Citations in this document: §1.3.

[174] Mike Hamburg, *OFFICIAL COMMENT: Round5 = Round2 + Hila5*, email dated 24 Aug 2018 11:30:24 -0700 (2018). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/YsGkKEJTt5c/m/V0eivEroAAAJ. Citations in this document: §1.

[175] Feng Hao, Sushmita Ruj, Sourav Sen Gupta (editors), *Progress in cryptology—INDOCRYPT 2019—20th international conference on cryptology in India, Hyderabad, India, December 15–18, 2019, proceedings*, Springer, 2019. ISBN 978-3-030-35422-0. URL: https://doi.org/10.1007/978-3-030-35423-7. See [100].

[176] Max Heiser, *Improved quantum hypercone locality sensitive filtering in lattice sieving* (2021). URL: https://eprint.iacr.org/2021/1295. Citations in this document: §1.

[177] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, *NTRU: a new high speed public key cryptosystem* (1996). URL: https://web.archive.org/web/20211027170520/https://ntru.org/f/hps96.pdf. Citations in this document: §1.3.

[178] Dennis Hofheinz, Alon Rosen (editors), *Theory of cryptography—17th international conference, TCC 2019, Nuremberg, Germany, December 1–5, 2019, proceedings, part II*, Lecture Notes in Computer Science, 11892, Springer, 2019. ISBN 978-3-030-36032-0. See [80].

[179] Thorsten Holz, Stefan Savage (editors), *25th USENIX security symposium, USENIX Security 16, Austin, TX, USA, August 10–12, 2016*, USENIX Association, 2016. URL: https://www.usenix.org/conference/usenixsecurity16. See [16].

[180] Iggy van Hoof, Elena Kirshanova, Alexander May, *Quantum key search for ternary LWE*, in PQCrypto 2021 [**114**] (2021), 117–132. URL: https://eprint.iacr.org/2021/865. Citations in this document: §1.

[181] Andreas Hülsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, *High-speed key encapsulation from NTRU*, in CHES 2017 [**148**] (2017), 232–252. URL: https://eprint.iacr.org/2017/667. Citations in this document: §2.2.

[182] Yuval Ishai, Vincent Rijmen (editors), *Advances in cryptology—EUROCRYPT 2019—38th annual international conference on the theory and applications of cryptographic techniques, Darmstadt, Germany, May 19–23, 2019, proceedings, part II*, Springer, 2019. ISBN 978-3-030-17655-6. See [311].

[183] Takanori Isobe, Toshihiro Ohigashi, Yuhei Watanabe, Masakatu Morii, *Full plaintext recovery attack on broadcast RC4*, in FSE 2013 [**272**] (2013), 179–202. URL: https://doi.org/10.1007/978-3-662-43933-3_10. Citations in this document: §1.3.

[184] Takanori Isobe, Toshihiro Ohigashi, Yuhei Watanabe, Masakatu Morii, *Comprehensive analysis of initial keystream biases of RC4*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E97.A** (2014), 139–151. URL: https://doi.org/10.1587/transfun.E97.A.139. Citations in this document: §1.3.

[185] Ryoma Ito, Atsuko Miyaji, *New integrated long-term glimpse of RC4*, in WISA 2014 [**317**] (2014), 137–149. URL: https://doi.org/10.1007/978-3-319-15087-1_11. Citations in this document: §1.3.

[186] Ryoma Ito, Atsuko Miyaji, *How TKIP induces biases of internal states of generic RC4*, in ACISP 2015 [**152**] (2015), 329–342. URL: https://doi.org/10.1007/978-3-319-19962-7_19. Citations in this document: §1.3.

[187] Ryoma Ito, Atsuko Miyaji, *Refined glimpse correlations of RC4*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E99.A** (2016), 3–13. URL: https://doi.org/10.1587/transfun.E99.A.3. Citations in this document: §1.3.

[188] Ryoma Ito, Atsuko Miyaji, *Refined RC4 key correlations of internal states in WPA*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E99.A** (2016), 1132–1144. URL: https://doi.org/10.1587/transfun.E99.A.1132. Citations in this document: §1.3.

[189] Ryoma Ito, Atsuko Miyaji, *New iterated RC4 key correlations*, in ACISP 2018 [**342**] (2018), 154–171. URL: https://doi.org/10.1007/978-3-319-93638-3_10. Citations in this document: §1.3.

[190] Ryoma Ito, Atsuko Miyaji, *New iterated RC4 key correlations and their application to plaintext recovery on WPA-TKIP*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E104.A** (2021), 190–202. URL: https://doi.org/10.1587/transfun.2020CIP0006. Citations in this document: §1.3.

[191] Sushil Jajodia, Chandan Mazumdar (editors), *Information systems security—7th international conference, ICISS 2011, Kolkata, India, December 15–19, 2011, procedings*, Springer, 2011. ISBN 978-3-642-25559-5. URL: https://doi.org/10.1007/978-3-642-25560-1. See [130].

[192] Amit Jana, Goutam Paul, *Revisiting RC4 key collision: faster search algorithm and new 22-byte colliding key pairs*, Cryptography and Communications **10** (2018), 479–508. URL: https://eprint.iacr.org/2016/1001. Citations in this document: §1.3.

[193] Sonu Jha, Subhadeep Banik, Takanori Isobe, Toshihiro Ohigashi, *Some proofs of joint distributions of keystream biases in RC4*, in Indocrypt 2016 [**142**] (2016), 305–321. URL: https://doi.org/10.1007/978-3-319-49890-4_17. Citations in this document: §1.3.

[194] Sonu Jha, Subhadeep Banik, Takanori Isobe, Toshihiro Ohigashi, Santanu Sarkar, *Theoretical understanding of some conditional and joint biases in RC4 stream cipher*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E101.A** (2018), 1869–1879. URL: https://doi.org/10.1587/transfun.E101.A.1869. Citations in this document: §1.3.

[195] Thomas Johansson, Subhamoy Maitra (editors), *Progress in cryptology—INDOCRYPT 2003, 4th international conference on cryptology in India, New Delhi, India, December 8–10, 2003, proceedings*, Springer, 2003. ISBN 3-540-20609-4. URL: https://doi.org/10.1007/b94593. See [300].

[196] Thomas Johansson, Phong Q. Nguyen (editors), *Advances in cryptology—EUROCRYPT 2013, 32nd annual international conference on the theory and applications of cryptographic techniques, Athens, Greece, May 26–30, 2013, proceedings*, Springer, 2013. ISBN 978-3-642-38347-2. See [157], [239].

[197] Antoine Joux (editor), *Fast software encryption—18th international workshop, FSE 2011, Lyngby, Denmark, February 13–16, 2011, revised selected papers*, Springer, 2011. ISBN 978-3-642-21701-2. URL: https://doi.org/10.1007/978-3-642-21702-9. See [245].

[198] Antoine Joux, Abderrahmane Nitaj, Tajjeeddine Rachidi (editors), *Progress in cryptology—AFRICACRYPT 2018—10th international conference on cryptology in Africa, Marrakesh, Morocco, May 7–9, 2018, proceedings*, Springer, 2018. ISBN 978-3-319-89338-9. URL: https://doi.org/10.1007/978-3-319-89339-6. See [64].

[199] Antoine Joux, Amr M. Youssef (editors), *Selected areas in cryptography—SAC 2014—21st international conference, Montreal, QC, Canada, August 14–15, 2014, revised selected papers*, Springer, 2014. ISBN 978-3-319-13050-7. See [145].

[200] Jaeyeon Jung, Thorsten Holz (editors), *24th USENIX security symposium, USENIX Security 15, Washington, D.C., USA, August 12–14, 2015*, USENIX Association, 2015. URL: https://www.usenix.org/conference/usenixsecurity15. See [158], [347].

[201] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, Ko Stoffelen, `pqm4`*: testing and benchmarking NIST PQC on ARM Cortex-M4* (2019). URL: https://eprint.iacr.org/2019/844. Citations in this document: §6.2.

[202] Shahram Khazaei, Willi Meier, *On reconstruction of RC4 keys from internal states*, in MMICS 2008 [**95**] (2008), 179–189. URL: https://doi.org/10.1007/978-3-540-89994-5_14. Citations in this document: §1.3.

[203] Aggelos Kiayias (editor), *Topics in cryptology—CT-RSA 2011—the cryptographers' track at the RSA Conference 2011, San Francisco, CA, USA, February 14–18, 2011, proceedings*, Lecture Notes in Computer Science, 6558, Springer, 2011. ISBN 978-3-642-19073-5. See [232].

[204] Yongdae Kim, Heejo Lee, Adrian Perrig (editors), *Information security applications—14th international workshop, WISA 2013, Jeju Island, Korea, August 19–21, 2013, revised selected papers*, Springer, 2014. ISBN 978-3-319-05148-2. URL: https://doi.org/10.1007/978-3-319-05149-9. See [234].

[205] Samuel T. King (editor), *Proceedings of the 22th USENIX security symposium, Washington, DC, USA, August 14–16, 2013*, USENIX Association, 2013. ISBN 978-1-931971-03-4. URL: https://www.usenix.org/conference/usenixsecurity13. See [14].

[206] Paul Kirchner, Pierre-Alain Fouque, *Revisiting lattice attacks on overstretched NTRU parameters*, in Eurocrypt 2017 [**121**] (2017), 3–26. URL: https://web.archive.org/web/20211010163907/https://www.di.ens.fr/~fouque/euro17a.pdf. Citations in this document: §3.13, §3.13, §3.13.

[207] Elena Kirshanova, Thijs Laarhoven, *Lower bounds on lattice sieving and information set decoding*, in Crypto 2021 [**247**] (2021), 791–820. URL: https://eprint.iacr.org/2021/785. Citations in this document: §1.

[208] Elena Kirshanova, Alexander May, *How to find ternary LWE keys using locality sensitive hashing* (2021). URL: https://eprint.iacr.org/2021/1255. Citations in this document: §1.

[209] Andreas Klein, *Attacks on the RC4 stream cipher*, Designs, Codes and Cryptography **48** (2008), 269–286. URL: https://doi.org/10.1007/s10623-008-9206-6. Citations in this document: §1.3.

[210] Lars R. Knudsen, Willi Meier, Bart Preneel, Vincent Rijmen, Sven Verdoolaege, *Analysis methods for (alleged) RC4*, in Asiacrypt 1998 [**289**] (1998), 327–341. URL: https://doi.org/10.1007/3-540-49649-1_26. Citations in this document: §1.3.

[211] Donald E. Knuth, *The art of computer programming, volume 3: sorting and searching*, 2nd edition, Addison-Wesley, Reading, 1998. ISBN 0-201-89685-0. Citations in this document: §B.

[212] Erik van der Kouwe, Dennis Andriesse, Herbert Bos, Cristiano Giuffrida, Gernot Heiser, *Benchmarking crimes: an emerging threat in systems security* (2018). URL: https://arxiv.org/abs/1801.02381. Citations in this document: §6, §6.1.

[213] Maurice Kraitchik, *Théorie des nombres*, Gauthier-Villars, 1922. Citations in this document: §2.3.

[214] Vlad Krasnov, *How "expensive" is crypto anyway?* (2017). URL: https://blog.cloudflare.com/how-expensive-is-crypto-anyway/. Citations in this document: §7.14, §7.14, §7.14.

[215] Robert Krauthgamer (editor), *Proceedings of the twenty-seventh annual ACM-SIAM symposium on discrete algorithms, SODA 2016, Arlington, VA, USA, January 10–12, 2016*, SIAM, 2016. ISBN 978-1-61197-433-1. See [78].

[216] Jin Kwak, Robert H. Deng, Yoojae Won, Guilin Wang (editors), *Information security, practice and experience, 6th international conference, ISPEC 2010, Seoul, Korea, May 12–13, 2010, proceedings*, Springer, 2010. ISBN 978-3-642-12826-4. URL: https://doi.org/10.1007/978-3-642-12827-1. See [107].

[217] Kris Kwiatkowski, *Towards post-quantum TLS* (2019). URL: https://eccworkshop.org/2019/slides/kwiatkowski.pdf. Citations in this document: §1.4, §6.6, §7.16, §A.3.

[218] Thijs Laarhoven, Artur Mariano, *Progressive lattice sieving*, in PQCrypto 2018 [**222**] (2018), 292–311. URL: https://eprint.iacr.org/2018/079. Citations in this document: §1.

[219] Xuejia Lai, Moti Yung, Dongdai Lin (editors), *Information security and cryptology—6th international conference, Inscrypt 2010, Shanghai, China, October 20–24, 2010, revised selected papers*, Springer, 2011. ISBN 978-3-642-21517-9. URL: https://doi.org/10.1007/978-3-642-21518-6. See [109].

[220] Xuejia Lai, Jianying Zhou, Hui Li (editors), *Information security, 14th international conference, ISC 2011, Xi'an, China, October 26–29, 2011, proceedings*, Springer, 2011. ISBN 978-3-642-24860-3. URL: https://doi.org/10.1007/978-3-642-24861-0. See [111].

[221] Tanja Lange, Kristin E. Lauter, Petr Lisonek (editors), *Selected areas in cryptography—SAC 2013—20th international conference, Burnaby, BC, Canada, August 14–16, 2013, revised selected papers*, Springer, 2014. ISBN 978-3-662-43413-0. URL: https://doi.org/10.1007/978-3-662-43414-7. See [286].

[222] Tanja Lange, Rainer Steinwandt (editors), *Post-quantum cryptography—9th international conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9–11, 2018, proceedings*, Springer, 2018. ISBN 978-3-319-79062-6. See [218].

[223] Adam Langley, *How to botch TLS forward secrecy* (2013). URL: https://www.imperialviolet.org/2013/06/27/botchingpfs.html. Citations in this document: §6.3.

[224] Adam Langley, *CECPQ1 results* (2016). URL: https://www.imperialviolet.org/2016/11/28/cecpq1.html. Citations in this document: §4.5, §6.6, §7.16.

[225] Adam Langley, *Post-quantum confidentiality for TLS* (2018). URL: https://www.imperialviolet.org/2018/04/11/pqconftls.html. Citations in this document: §1.4.

[226] Adam Langley, *CECPQ2* (2018). URL: https://www.imperialviolet.org/2018/12/12/cecpq2.html. Citations in this document: §A.1.

[227] Adeline Langlois, Damien Stehlé, *Worst-case to average-case reductions for module lattices*, Designs, Codes and Cryptography **75** (2015), 565–599. URL: https://eprint.iacr.org/2012/090. Citations in this document: §5.4.

[228] Arjen K. Lenstra, Hendrik W. Lenstra, Jr. (editors), *The development of the number field sieve*, Lecture Notes in Mathematics, 1554, Springer-Verlag, 1993. ISBN 3-540-57013-6. See [94].

[229] Andrea Lesavourey, Thomas Plantard, Willy Susilo, *Short principal ideal problem in multicubic fields*, Journal of Mathematical Cryptology **14** (2020), 359–392. Citations in this document: §2.5.

[230] Pierre-Yvan Liardet, Nele Mentens (editors), *Smart card research and advanced applications—19th international conference, CARDIS 2020, virtual event, November 18–19, 2020, revised selected papers*, Springer, 2021. ISBN 978-3-030-68486-0. See [251].

[231] Jay Ligatti, Xinming Ou, Jonathan Katz, Giovanni Vigna (editors), *CCS '20: 2020 ACM SIGSAC conference on computer and communications security, virtual event, USA, November 9–13, 2020*, ACM, 2020. ISBN 978-1-4503-7089-9. URL: https://doi.org/10.1145/3372297. See [333].

[232] Richard Lindner, Chris Peikert, *Better key sizes (and attacks) for LWE-based encryption*, in CT-RSA [**203**] (2011), 319–339. URL: https://eprint.iacr.org/2010/613. Citations in this document: §1.3, §1.3.

[233] Claudio L. Lucchesi, Arnaldo V. Moura (editors), *LATIN'98: theoretical informatics. Proceedings of the 3rd Latin American symposium held in Campinas, April 20–24, 1998*, Lecture Notes in Computer Science, 1380, Springer, 1998. ISBN ISBN 3-540-64275-7. See [90].

[234] Jing Lv, Bin Zhang, Dongdai Lin, *Some new weaknesses in the RC4 stream cipher*, in WISA 2013 [**204**] (2013), 28–38. URL: https://doi.org/10.1007/978-3-319-05149-9_2. Citations in this document: §1.3.

[235] Vadim Lyubashevsky, *OFFICIAL COMMENT: CRYSTALS-DILITHIUM*, email dated 2 Jan 2018 17:34:08 +0100 (2018). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/aWxC2ynJDLE/m/YOsMJ2ewAAAJ. Citations in this document: §3.6.

[236] Vadim Lyubashevsky, *Re: Non-applicability of the CNRS patent to Kyber and Saber, and next steps*, email dated 11 Oct 2021 12:11:39 +0200 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/2Xv0mrF9lVo/m/pb5tOwqrAAAJ. Citations in this document: §5.9.

[237] Vadim Lyubashevsky, *Re: Kyber's inefficiency: some data points*, email dated 26 Oct 2021 11:43:26 +0200 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/ik1pXoAZk8s/m/icfps2elAgAJ. Citations in this document: §7.8, §7.8.

[238] Vadim Lyubashevsky, Chris Peikert, Oded Regev, *On ideal lattices and learning with errors over rings*, Journal of the ACM **60** (2010), 43:1–43:35. URL: https://eprint.iacr.org/2012/230. Citations in this document: §5.9.

[239] Vadim Lyubashevsky, Chris Peikert, Oded Regev, *A toolkit for Ring-LWE cryptography*, in Eurocrypt 2013 [**196**] (2013), 35–54. URL: https://eprint.iacr.org/2013/293. Citations in this document: §6.7.

[240] Vadim Lyubashevsky, Damien Stehlé, *Non-applicability of the CNRS patent to Kyber and Saber, and next steps*, email dated 8 Oct 2021 21:33:23 +0200 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/2Xv0mrF9lVo/m/e0oKQhIIBwAJ. Citations in this document: §3.17, §5.8, §6.2, §6.2.

[241] Vadim Lyubashevsky, Damien Stehlé, *Non-applicability of the Gaborit&Aguilar-Melchor patent to Kyber and Saber* (2021). URL: https://eprint.iacr.org/2021/1364. Citations in this document: §3.17.

[242] Subhamoy Maitra, Sourav Sen Gupta, *New long-term glimpse of RC4 stream cipher*, in ICISS 2013 [**30**] (2013), 230–238. URL: https://doi.org/10.1007/978-3-642-45204-8_17. Citations in this document: §1.3.

[243] Subhamoy Maitra, Goutam Paul, *Analysis of RC4 and proposal of additional layers for better security margin*, in Indocrypt 2008 [**115**] (2008), 27–39. URL: https://eprint.iacr.org/2008/396. Citations in this document: §1.3.

[244] Subhamoy Maitra, Goutam Paul, *New form of permutation bias and secret key leakage in keystream bytes of RC4*, in FSE 2008 [**285**] (2008), 253–269. URL: https://eprint.iacr.org/2007/261. Citations in this document: §1.3.

[245] Subhamoy Maitra, Goutam Paul, Sourav Sen Gupta, *Attack on broadcast RC4 revisited*, in FSE 2011 [**197**] (2011), 199–217. URL: https://doi.org/10.1007/978-3-642-21702-9_12. Citations in this document: §1.3.

[246] Subhamoy Maitra, Goutam Paul, Santanu Sarkar, Michael Lehmann, Willi Meier, *New results on generalization of Roos-type biases and related keystreams of RC4*, in Africacrypt 2013 [**359**] (2013), 222–239. URL: https://doi.org/10.1007/978-3-642-38553-7_13. Citations in this document: §1.3.

[247] Tal Malkin, Chris Peikert (editors), *Advances in cryptology—CRYPTO 2021—41st annual international cryptology conference, CRYPTO 2021, virtual event, August 16–20, 2021, proceedings, part II*, Springer, 2021. ISBN 978-3-030-84244-4. See [10], [207], [257].

[248] Itsik Mantin, *A practical attack on the fixed RC4 in the WEP mode*, in Asiacrypt 2005 [**318**] (2005), 395–411. URL: https://doi.org/10.1007/11593447_21. Citations in this document: §1.3.
[249] Itsik Mantin, *Predicting and distinguishing attacks on RC4 keystream generator*, in Eurocrypt 2005 [**122**] (2005), 491–506. URL: https://doi.org/10.1007/11426639_29. Citations in this document: §1.3.
[250] Itsik Mantin, Adi Shamir, *A practical attack on broadcast RC4*, in FSE 2001 [**253**] (2001), 152–164. URL: https://doi.org/10.1007/3-540-45473-X_13. Citations in this document: §1.3.
[251] Adrian Marotzke, *A constant time full hardware implementation of Streamlined NTRU Prime*, in CARDIS 2020 [**230**] (2020), 3–17. URL: https://eprint.iacr.org/2020/1067. Citations in this document: §7.10.
[252] John F. Martin, *The myth of the 18-month delay in publishing patent applications* (2015). URL: https://web.archive.org/web/20210930201215/https://www.ipwatchdog.com/2015/08/03/the-myth-of-the-18-month-delay-in-publishing-patent-applications/id=60185/. Citations in this document: §3.18.
[253] Mitsuru Matsui (editor), *Fast software encryption, 8th international workshop, FSE 2001, Yokohama, Japan, April 2–4, 2001, revised papers*, Springer, 2002. ISBN 3-540-43869-6. URL: https://doi.org/10.1007/3-540-45473-X. See [250].
[254] Mitsuru Matsui, *Key collisions of the RC4 stream cipher*, in FSE 2009 [**141**] (2009), 38–50. URL: https://doi.org/10.1007/978-3-642-03317-9_3. Citations in this document: §1.3.
[255] Alexander Maximov, Dmitry Khovratovich, *New state recovery attack on RC4*, in Crypto 2008 [**350**] (2008), 297–316. URL: https://eprint.iacr.org/2008/017. Citations in this document: §1.3.
[256] Greg Maxwell, *Re: Patent update*, email dated 25 Oct 2021 19:52:24 +0000 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/RSv2bVOV10U/m/g8VbYgh4AgAJ. Citations in this document: §4.3.
[257] Alexander May, *How to meet ternary LWE keys*, in Crypto 2021 [**247**] (2021), 701–731. URL: https://eprint.iacr.org/2021/216. Citations in this document: §1.
[258] David McGrew, *Living with postquantum cryptography* (2015). URL: https://csrc.nist.gov/csrc/media/events/workshop-on-cybersecurity-in-a-post-quantum-world/documents/presentations/session4-mcgrew-david.pdf. Citations in this document: §3.1, §A.2.
[259] Daniele Micciancio, Oded Regev, *Lattice-based cryptography*, in Post-quantum cryptography [**67**] (2009), 147–191. URL: https://web.archive.org/web/20210506211055/https://cims.nyu.edu/~regev/papers/pqc.pdf. Citations in this document: §5.9.
[260] Daniele Micciancio, Thomas Ristenpart (editors), *Advances in cryptology—CRYPTO 2020—40th annual international cryptology conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, proceedings, part II*, Springer, 2020. ISBN 978-3-030-56879-5. See [9], [128], [169].
[261] Victor S. Miller, *Use of elliptic curves in cryptography*, in Crypto 1985 [**357**] (1986), 417–426. URL: https://link.springer.com/content/pdf/10.1007/3-540-39799-X_31.pdf. Citations in this document: §2.3.
[262] Ali Miri, Serge Vaudenay (editors), *Selected areas in cryptography—18th international workshop, SAC 2011, Toronto, ON, Canada, August 11–12, 2011,*

revised selected papers, Springer, 2012. ISBN 978-3-642-28495-3. URL: https://doi.org/10.1007/978-3-642-28496-0. See [172].

[263] Ilya Mironov, *(Not so) random shuffles of RC4*, in Crypto 2002 [**360**] (2002), 304–319. URL: https://doi.org/10.1007/3-540-45708-9_20. Citations in this document: §1.3.

[264] Serge Mister, Stafford E. Tavares, *Cryptanalysis of RC4-like ciphers*, in SAC 1998 [**343**] (1998), 131–143. URL: https://doi.org/10.1007/3-540-48892-8_11. Citations in this document: §1.3.

[265] Michael Mitzenmacher (editor), *Proceedings of the 41st annual ACM symposium on theory of computing, STOC 2009, Bethesda, MD, USA, May 31–June 2, 2009*, ACM, 2009. ISBN 978-1-60558-506-2. See [161].

[266] Atsuko Miyaji, Masahiro Sukegawa, *New correlations of RC4 PRGA using nonzero-bit differences*, in ACISP 2009 [**86**] (2009), 134–152. URL: https://doi.org/10.1007/978-3-642-02620-1_10. Citations in this document: §1.3.

[267] Atsuko Miyaji, Masahiro Sukegawa, *New analysis based on correlations of RC4 PRGA with nonzero-bit differences*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E93.A** (2010), 1066–1077. URL: https://doi.org/10.1587/transfun.E93.A.1066. Citations in this document: §1.3.

[268] Dustin Moody, *Signed Statements*, email dated 8 May 2018 13:35:17 +0000 (2018). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/y-rUAw-V8VM/m/8-i-okdSAgAJ. Citations in this document: §3.17.

[269] Dustin Moody, *Re: ROUND 3 OFFICIAL COMMENT: CRYSTALS-KYBER*, email dated 8 Dec 2020 13:11:49 -0800 (2020). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/NSe0wAzKJtA/m/WqWmlCiUAwAJ. Citations in this document: §1.3.

[270] Dustin Moody, Yi-Kai Liu, Angela Robinson, Ray Perlner, Daniel Apon, Daniel Smith-Tone, *NIST Q&A* (2021). URL: https://www.nist.gov/video/third-pqc-standardization-conference-session-ix-implementations-ii-and-nist-qa. Citations in this document: §3.17.

[271] Dustin Moody, Ray Perlner, *Re: ROUND 2 OFFICIAL COMMENT: NTRU Prime*, email dated 28 Sep 2020 21:15:08 +0000 (2020). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/vVepj-wQkYs/m/aWXdSevyBgAJ. Citations in this document: §3.5.

[272] Shiho Moriai (editor), *Fast software encryption—20th international workshop, FSE 2013, Singapore, March 11–13, 2013, revised selected papers*, Springer, 2014. ISBN 978-3-662-43932-6. URL: https://doi.org/10.1007/978-3-662-43933-3. See [183].

[273] Shiho Moriai, Huaxiong Wang (editors), *Advances in cryptology—ASIACRYPT 2020—26th international conference on the theory and application of cryptology and information security, Daejeon, South Korea, December 7–11, 2020, proceedings, part II*, Springer, 2020. ISBN 978-3-030-64833-6. See [40].

[274] Masakatu Morii, Yosuke Todo, *Cryptanalysis for RC4 and breaking WEP/WPA-TKIP*, IEICE Transactions on Information and Systems **E94.D** (2011), 2087–2094. URL: https://doi.org/10.1587/transinf.E94.D.2087. Citations in this document: §1.3.

[275] Michele Mosca (editor), *Post-quantum cryptography—6th international workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1–3, 2014, proceedings*, Springer, 2014. ISBN 978-3-319-11658-7. See [302].

[276] Atsushi Nagao, Toshihiro Ohigashi, Takanori Isobe, Masakatu Morii, *Expanding weak-key space of RC4*, Journal of Information Processing **22** (2014), 357–365. URL: https://doi.org/10.2197/ipsjjip.22.357. Citations in this document: §1.3.

[277] National Institute of Standards and Technology, *Announcing request for nominations for public-key post-quantum cryptographic algorithms*, Federal Register **81** (2016), 92787–92788. URL: https://www.federalregister.gov/documents/2016/12/20/2016-30615/announcing-request-for-nominations-for-public-key-post-quantum-cryptographic-algorithms. Citations in this document: §A.

[278] National Institute of Standards and Technology, *Proposed submission requirements and evaluation criteria for the post-quantum cryptography standardization process* (2016). URL: https://web.archive.org/web/20210321052818/https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-draft-aug-2016.pdf. Citations in this document: §B.1, §B.2, §B.5.

[279] National Institute of Standards and Technology, *Submission requirements and evaluation criteria for the post-quantum cryptography standardization process* (2016). URL: https://web.archive.org/web/20211023071507/https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf. Citations in this document: §4.3, §4.3, §A, §A.1, §A.2, §A.3, §B.2, §B.2, §B.5.

[280] Kalle Ngo, Elena Dubrova, Qian Guo, Thomas Johansson, *A side-channel attack on a masked IND-CCA secure SABER KEM implementation*, IACR Transactions on Cryptographic Hardware and Embedded Systems **2021.4** (2021), 676–707. URL: https://eprint.iacr.org/2021/079. Citations in this document: §7.10, §A.1.

[281] Phong Q. Nguyen, Elisabeth Oswald (editors), *Advances in cryptology—EUROCRYPT 2014—33rd annual international conference on the theory and applications of cryptographic techniques, Copenhagen, Denmark, May 11–15, 2014, proceedings*, Lecture Notes in Computer Science, 8441, Springer, 2014. ISBN 978-3-642-55219-9. See [33].

[282] Phong Q. Nguyen, David Pointcheval (editors), *Public Key Cryptography—PKC 2010, 13th international conference on practice and theory in public key cryptography, Paris, France, May 26–28, 2010, proceedings*, Springer, 2010. ISBN 978-3-642-13012-0. See [339].

[283] Abderrahmane Nitaj, Amr M. Youssef (editors), *Progress in cryptology—AFRICACRYPT 2020—12th international conference on cryptology in Africa, Cairo, Egypt, July 20–22, 2020, proceedings*, Springer, 2020. ISBN 978-3-030-51937-7. See [136].

[284] NTRU Prime Risk-Management Team, *NTRU Prime: Warnings* (2021). URL: https://ntruprime.cr.yp.to/warnings.html. Citations in this document: §A.3.

[285] Kaisa Nyberg (editor), *Fast software encryption, 15th international workshop, FSE 2008, Lausanne, Switzerland, February 10–13, 2008, revised selected papers*, Springer, 2008. ISBN 978-3-540-71038-7. URL: https://doi.org/10.1007/978-3-540-71039-4. See [79], [244].

[286] Toshihiro Ohigashi, Takanori Isobe, Yuhei Watanabe, Masakatu Morii, *How to recover any byte of plaintext on RC4*, in SAC 2013 [**221**] (2013), 155–173. URL:

https://doi.org/10.1007/978-3-662-43414-7_8. Citations in this document: §1.3.

[287] Toshihiro Ohigashi, Takanori Isobe, Yuhei Watanabe, Masakatu Morii, *Full plaintext recovery attacks on RC4 using multiple biases*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E98.A** (2015), 81–91. URL: https://doi.org/10.1587/transfun.E98.A.81. Citations in this document: §1.3.

[288] Toshihiro Ohigashi, Yoshiaki Shiraishi, Masakatu Morii, *New weakness in the key-scheduling algorithm of RC4*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E91.A** (2008), 3–11. URL: https://doi.org/10.1093/ietfec/e91-a.1.3. Citations in this document: §1.3.

[289] Kazuo Ohta, Dingyi Pei (editors), *Advances in cryptology—ASIACRYPT '98, international conference on the theory and applications of cryptology and information security, Beijing, China, October 18–22, 1998, proceedings*, Springer, 1998. ISBN 3-540-65109-8. URL: https://doi.org/10.1007/3-540-49649-1. See [210].

[290] OpenBSD Project, *Add experimental post-quantum hybrid key exchange method based on Streamlined NTRU Prime (coupled with X25519)* (2021). URL: https://web.archive.org/web/20211027175935/https://cvsweb.openbsd.org/src/sbin/iked/dh.h?rev=1.15&content-type=text%2Fx-cvsweb-markup. Citations in this document: §A.3.

[291] OpenSSH Project, *OpenSSH release notes* (2021). URL: https://www.openssh.com/releasenotes.html. Citations in this document: §A.3.

[292] Kenneth G. Paterson (editor), *Advances in cryptology—EUROCRYPT 2011—30th annual international conference on the theory and applications of cryptographic techniques, Tallinn, Estonia, May 15–19, 2011, proceedings*, Springer, 2011. ISBN 978-3-642-20464-7. URL: https://doi.org/10.1007/978-3-642-20465-4. See [162], [335], [341].

[293] Kenneth G. Paterson, Bertram Poettering, Jacob C. N. Schuldt, *Big bias hunting in Amazonia: large-scale computation and exploitation of RC4 biases (invited paper)*, in Asiacrypt 2014 [**325**] (2014), 398–419. URL: https://doi.org/10.1007/978-3-662-45611-8_21. Citations in this document: §1.3.

[294] Kenneth G. Paterson, Jacob C. N. Schuldt, *Statistical attacks on cookie masking for RC4*, Cryptography and Communications **10** (2018), 777–801. URL: https://eprint.iacr.org/2018/093. Citations in this document: §1.3.

[295] Goutam Paul, Subhamoy Maitra, *Permutation after RC4 key scheduling reveals the secret key*, in SAC 2007 [**5**] (2007), 360–377. URL: https://doi.org/10.1007/978-3-540-77360-3_23. Citations in this document: §1.3.

[296] Goutam Paul, Subhamoy Maitra, *On biases of permutation and keystream bytes of RC4 towards the secret key*, Cryptography and Communications **1** (2009), 225–268. URL: https://doi.org/10.1007/s12095-008-0009-4. Citations in this document: §1.3.

[297] Goutam Paul, Subhamoy Maitra, Rohit Srivastava, *On non-randomness of the permutation after RC4 key scheduling*, in AAECC 2007 [**88**] (2007), 100–109. URL: https://eprint.iacr.org/2007/305. Citations in this document: §1.3.

[298] Goutam Paul, Siddheshwar Rathi, Subhamoy Maitra, *On non-negligible bias of the first output byte of RC4 towards the first three bytes of the secret key*, Designs, Codes and Cryptography **49** (2008), 123–134. URL: https://doi.org/10.1007/s10623-008-9177-7. Citations in this document: §1.3.

[299] Goutam Paul, Souvik Ray, *Analysis of burn-in period for RC4 state transition*, Cryptography and Communications **10** (2018), 881–908. URL: https://eprint.iacr.org/2017/175. Citations in this document: §1.3.

[300] Souradyuti Paul, Bart Preneel, *Analysis of non-fortuitous predictive states of the RC4 keystream generator*, in INDOCRYPT 2003 [**195**] (2003), 52–67. URL: https://doi.org/10.1007/978-3-540-24582-7_4. Citations in this document: §1.3.

[301] Souradyuti Paul, Bart Preneel, *A new weakness in the RC4 keystream generator and an approach to improve the security of the cipher*, in FSE 2004 [**319**] (2004), 245–259. URL: https://doi.org/10.1007/978-3-540-25937-4_16. Citations in this document: §1.3.

[302] Chris Peikert, *Lattice cryptography for the Internet*, in PQCrypto 2014 [**275**] (2014), 197–219. URL: https://eprint.iacr.org/2014/070. Citations in this document: §3.17.

[303] Chris Peikert, *Ideal lattices and Ring-LWE: overview and open problems* (2015). URL: https://web.archive.org/web/20150923212549/https://icerm.brown.edu/materials/Slides/tw-15-7/Ideal_Lattices_and_Ring-LWE-_Overview_and_Open_Problems_]_Chris_Peikert,_Georgia_Institute_of_Technology___.pdf. Citations in this document: §5.6.

[304] Chris Peikert, *Finding short generators of ideals, and implications for cryptography* (2016). URL: https://web.archive.org/web/20210818101136/https://www.mathematik.uni-kl.de/~thofmann/ants/talks/mon/talk_peikert.pdf. Citations in this document: §1.2.

[305] Chris Peikert, *A decade of lattice cryptography* (2016). URL: https://eprint.iacr.org/2015/939. Citations in this document: §5.1, §5.9.

[306] Chris Peikert, *Lattice-based cryptography* (2016). URL: https://www.youtube.com/watch?v=FVFw_qb1ZkY. Citations in this document: §1.2.

[307] Chris Peikert, *Lattice-based cryptography* (2016). URL: https://web.archive.org/web/20210628010013/https://web.eecs.umich.edu/~cpeikert/pubs/slides-qcrypt.pdf. Citations in this document: §1.2.

[308] Chris Peikert, *Re: qrom security etc.*, email dated 2 Dec 2017 11:12:56 -0500 (2017). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/WxRmVPhAENw/m/r_vtIbRMAQAJ. Citations in this document: §5.9.

[309] Chris Peikert, *The state of lattice-based assumptions* (2018). URL: https://www.youtube.com/watch?v=VUGHJipRfwA. Citations in this document: §1.2.

[310] Chris Peikert, *Re: ROUND 3 OFFICIAL COMMENT: CRYSTALS-KYBER*, email dated 14 Dec 2020 11:52:25 -0500 (2020). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/NSe0wAzKJtA/m/DxGrdfWHAgAJ. Citations in this document: §3.17.

[311] Alice Pellet-Mary, Guillaume Hanrot, Damien Stehlé, *Approx-SVP in ideal lattices with pre-processing*, in [**182**] (2019), 685–716. URL: https://eprint.iacr.org/2019/215. Citations in this document: §1, §1, §1.2, §1.2, §1.2, §1.2.

[312] Bo-Yuan Peng, Adrian Marotzke, Ming-Han Tsai, Bo-Yin Yang, Ho-Lin Chen, *Streamlined NTRU Prime on FPGA* (2021). URL: https://eprint.iacr.org/2021/1444. Citations in this document: §7.10, §7.10.

[313] Thomas Peyrin, Steven D. Galbraith (editors), *Advances in cryptology—ASIACRYPT 2018—24th international conference on the theory and application of cryptology and information security, Brisbane, QLD, Australia, December 2–6, 2018, proceedings, part I*, Springer, 2018. ISBN 978-3-030-03325-5. See [18], [31].

[314] Raphael C.-W. Phan, Moti Yung (editors), *Paradigms in cryptology—Mycrypt 2016. Malicious and exploratory cryptology—second international conference, Mycrypt 2016, Kuala Lumpur, Malaysia, December 1–2, 2016, revised selected papers*, Lecture Notes in Computer Science, 10311, Springer, 2017. ISBN 978-3-319-61272-0. See [104].

[315] David Pointcheval, Thomas Johansson (editors), *Advances in cryptology—EUROCRYPT 2012—31st annual international conference on the theory and applications of cryptographic techniques, Cambridge, UK, April 15–19, 2012, proceedings*, Springer, 2012. ISBN 978-3-642-29010-7. See [32].

[316] Thomas Pornin, *OFFICIAL COMMENT: Falcon (bug & fixes)*, email dated 18 Sep 2019 07:17:36 -0700 (2019). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/7Z8x5AMXy8s/m/Spyv8VYoBQAJ. Citations in this document: §3.6.

[317] Kyung Hyune Rhee, Jeong Hyun Yi (editors), *Information security applications—15th international workshop, WISA 2014, Jeju Island, Korea, August 25–27, 2014, revised selected papers*, Springer, 2015. ISBN 978-3-319-15086-4. URL: https://doi.org/10.1007/978-3-319-15087-1. See [185].

[318] Bimal K. Roy (editor), *Advances in cryptology—ASIACRYPT 2005, 11th international conference on the theory and application of cryptology and information security, Chennai, India, December 4–8, 2005, proceedings*, Springer, 2005. ISBN 3-540-30684-6. URL: https://doi.org/10.1007/11593447. See [248].

[319] Bimal K. Roy, Willi Meier (editors), *Fast software encryption, 11th international workshop, FSE 2004, Delhi, India, February 5–7, 2004, revised papers*, Springer, 2004. ISBN 3-540-22171-9. URL: https://doi.org/10.1007/b98177. See [301].

[320] Sujoy Sinha Roy, Andrea Basso, *High-speed instruction-set coprocessor for lattice-based key encapsulation mechanism: SABER in hardware*, IACR Transactions on Cryptographic Hardware and Embedded Systems **2020** (2020), 443–466. URL: https://eprint.iacr.org/2020/434. Citations in this document: §6.5.

[321] Markku Saarinen, *ROUND 3 OFFICIAL COMMENT: FrodoKEM – CCA Bug*, email dated 10 Dec 2020 07:11:18 -0800 (2020). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/kSUKzDNc5ME/m/EMFYz9RNCAAJ. Citations in this document: §A.3.

[322] Markku Saarinen, *Re: Kyber's inefficiency: some data points*, email dated 23 Oct 2021 16:32:51 +0100 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/ik1pXoAZk8s/m/sKfmfrXMAQAJ. Citations in this document: §7.8.

[323] Markku Saarinen, *Re: Patent update*, email dated 25 Oct 2021 20:31:55 +0100 (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/RSv2bVOV10U/m/akFUaep2AgAJ. Citations in this document: §4.3.

[324] Ahmad-Reza Sadeghi, Virgil D. Gligor, Moti Yung (editors), *2013 ACM SIGSAC conference on computer and communications security, CCS'13, Berlin, Germany, November 4–8, 2013*, ACM, 2013. ISBN 978-1-4503-2477-9. See [70].

[325] Palash Sarkar, Tetsu Iwata (editors), *Advances in cryptology—ASIACRYPT 2014—20th international conference on the theory and application of cryptology and information security, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014, proceedings, part I*, Springer, 2014. ISBN 978-3-662-45610-1. URL: https://doi.org/10.1007/978-3-662-45611-8. See [293].

[326] Santanu Sarkar, *Proving empirical key-correlations in RC4*, Information Processing Letters **114** (2014), 234–238. URL: https://doi.org/10.1016/j.ipl.2014.01.002. Citations in this document: §1.3.

[327] Santanu Sarkar, *Further non-randomness in RC4, RC4A and VMPC*, Cryptography and Communications **7** (2015), 317–330. URL: https://doi.org/10.1007/s12095-014-0119-0. Citations in this document: §1.3.

[328] Santanu Sarkar, *Results on significant anomalies of state values after key scheduling algorithm in RC4*, IET Information Security **11** (2017), 267–272. URL: https://doi.org/10.1049/iet-ifs.2016.0451. Citations in this document: §1.3.

[329] Santanu Sarkar, Sourav Sen Gupta, Goutam Paul, Subhamoy Maitra, *Proving TLS-attack related open biases of RC4*, Designs, Codes and Cryptography **77** (2015), 231–253. URL: https://eprint.iacr.org/2013/502. Citations in this document: §1.3.

[330] Santanu Sarkar, Ayineedi Venkateswarlu, *Revisiting (nested) Roos bias in RC4 key scheduling algorithm*, Designs, Codes and Cryptography (2017), 131–148. URL: https://doi.org/10.1007/s10623-016-0219-2. Citations in this document: §1.3.

[331] Bruce Schneier (editor), *Fast software encryption, 7th international workshop, FSE 2000, New York, NY, USA, April 10–12, 2000, proceedings*, Springer, 2001. ISBN 3-540-41728-1. URL: https://doi.org/10.1007/3-540-44706-7. See [151].

[332] Marc Schoolderman, Jonathan Moerman, Sjaak Smetsers, Marko C. J. D. van Eekelen, *Efficient verification of optimized code: correct high-speed X25519*, in NFM 2021 [**143**] (2021), 304–321. URL: https://eprint.iacr.org/2021/415. Citations in this document: §7.14.

[333] Peter Schwabe, Douglas Stebila, Thom Wiggers, *Post-quantum TLS without handshake signatures*, in CCS 2020 [**231**] (2020), 1461–1480. URL: https://eprint.iacr.org/2020/534. Citations in this document: §7.16.

[334] Pouyan Sepehrdad, Serge Vaudenay, Martin Vuagnoux, *Discovery and exploitation of new biases in RC4*, in SAC 2010 [**81**] (2010), 74–91. URL: https://doi.org/10.1007/978-3-642-19574-7_5. Citations in this document: §1.3.

[335] Pouyan Sepehrdad, Serge Vaudenay, Martin Vuagnoux, *Statistical attack on RC4—distinguishing WPA*, in Eurocrypt 2011 [**292**] (2011), 343–363. URL: https://doi.org/10.1007/978-3-642-20465-4_20. Citations in this document: §1.3.

[336] David B. Shmoys (editor), *Symposium on theory of computing, STOC 2014, New York, NY, USA, May 31–June 03, 2014*, ACM, 2014. ISBN 978-1-4503-2710-7. See [144].

[337] Joseph H. Silverman, *An introduction to the theory of lattices and applications to cryptography* (2006). URL: https://web.archive.org/web/20211027172828/https://www.math.brown.edu/johsilve/Presentations/WyomingLattices.pdf. Citations in this document: §1.3.

[338] Joseph H. Silverman, Nigel P. Smart, Frederik Vercauteren, *An algebraic approach to NTRU ($q = 2^n$) via Witt vectors and overdetermined systems of nonlinear equations*, in SCN 2004 [**83**] (2004), 278–293. Citations in this document: §3.12.

[339] Nigel P. Smart, Frederik Vercauteren, *Fully homomorphic encryption with relatively small key and ciphertext sizes*, in PKC 2010 [**282**] (2010), 420–443. URL: https://eprint.iacr.org/2009/571. Citations in this document: §1.2.

[340] Douglas Stebila, Michele Mosca, *Post-quantum key exchange for the Internet and the Open Quantum Safe project*, in SAC 2016 [**27**] (2016), 14–37. URL: https://eprint.iacr.org/2016/1017. Citations in this document: §A.1.

[341] Damien Stehlé, Ron Steinfeld, *Making NTRU as secure as worst-case problems over ideal lattices*, in Eurocrypt 2011 [**292**] (2011), 27–47. URL: https://web.archive.org/web/20181222161831/http://perso.ens-lyon.fr/damien.stehle/downloads/ntruenc.pdf. Citations in this document: §5.3, §5.3, §5.3.

[342] Willy Susilo, Guomin Yang (editors), *Information security and privacy—23rd Australasian conference, ACISP 2018, Wollongong, NSW, Australia, July 11–13, 2018, proceedings*, Springer, 2018. ISBN 978-3-319-93637-6. URL: https://doi.org/10.1007/978-3-319-93638-3. See [189].

[343] Stafford E. Tavares, Henk Meijer (editors), *Selected areas in cryptography '98, SAC'98, Kingston, Ontario, Canada, August 17–18, 1998, proceedings*, Springer, 1999. ISBN 3-540-65894-7. URL: https://doi.org/10.1007/3-540-48892-8. See [264].

[344] Ryoichi Teramura, Toshihiro Ohigashi, Hidenori Kuwakado, Masakatu Morii, *Generalized classes of weak keys on RC4 using predictive state*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E94.A** (2011), 10–18. URL: https://doi.org/10.1587/transfun.E94.A.10. Citations in this document: §1.3.

[345] United States Court of Appeals for the Federal Circuit, *Crown Packaging Tech., Inc. v. Rexam Beverage Can Co.*, 559 F.3d 1308 (2009). URL: https://law.justia.com/cases/federal/appellate-courts/cafc/08-1284/08-1284-2011-03-27.html. Citations in this document: §3.17.

[346] United States Supreme Court, *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 535 U.S. 722 (2002). URL: https://tile.loc.gov/storage-services/service/ll/usrep/usrep535/usrep535722/usrep535722.pdf. Citations in this document: §3.17.

[347] Mathy Vanhoef, Frank Piessens, *All your biases belong to us: breaking RC4 in WPA-TKIP and TLS*, in USENIX Security 2015 [**200**] (2015), 97–112. URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/vanhoef. Citations in this document: §1.3.

[348] Serge Vaudenay, Martin Vuagnoux, *Passive-only key recovery attacks on RC4*, in SAC 2007 [**5**] (2007), 344–359. URL: https://doi.org/10.1007/978-3-540-77360-3_22. Citations in this document: §1.3.

[349] Serge Vaudenay, Amr M. Youssef (editors), *Selected areas in cryptography, 8th annual international workshop, SAC 2001 Toronto, Ontario, Canada, August 16–17, 2001, revised papers*, Springer, 2001. ISBN 3-540-43066-0. URL: https://doi.org/10.1007/3-540-45537-X. See [150].

[350] David A. Wagner (editor), *Advances in cryptology—CRYPTO 2008, 28th annual international cryptology conference, Santa Barbara, CA, USA, August 17–21, 2008, proceedings*, Springer, 2008. ISBN 978-3-540-85173-8. URL: https://doi.org/10.1007/978-3-540-85174-5. See [255].

[351] Chia C. Wang, Kimberly A. Prather, Josué Sznitman, Jose L. Jimenez, Seema S. Lakdawala, Zeynep Tufekci, Linsey C. Marr, *Airborne transmission of respiratory viruses*, Science **373.6558** (2021), document eabd9149. URL: https://www.science.org/doi/abs/10.1126/science.abd9149. Citations in this document: §3.5.

[352] Lawrence C. Washington, *Introduction to cyclotomic fields* (1997). ISBN 0-387-94762-0. Citations in this document: §2.6, §2.6.

[353] Yuhei Watanabe, Takanori Isobe, Toshihiro Ohigashi, Masakatu Morii, *How to efficiently exploit different types of biases for plaintext recovery of RC4*, IEICE

Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E100.A** (2017), 803–810. URL: https://doi.org/10.1587/transfun.E100.A.803. Citations in this document: §1.3.

[354] Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, Shai Halevi (editors), *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, Vienna, Austria, October 24–28, 2016*, ACM, 2016. ISBN 978-1-4503-4139-4. URL: https://dl.acm.org/citation.cfm?id=2976749. See [85].

[355] A. E. Western, J. C. P. Miller, *Tables of indices and primitive roots*, Cambridge University Press, 1968. Citations in this document: §2.3.

[356] Michael J. Wiener (editor), *Advances in cryptology—CRYPTO '99, 19th annual international cryptology conference, Santa Barbara, California, USA, August 15–19, 1999, proceedings*, Lecture Notes in Computer Science, 1666, Springer, 1999. ISBN 3-540-66347-9. See [153].

[357] Hugh C. Williams (editor), *Advances in cryptology—CRYPTO '85, Santa Barbara, California, USA, August 18–22, 1985, proceedings*, Lecture Notes in Computer Science, 218, Springer, 1986. ISBN 3-540-16463-4. See [261].

[358] Kenneth Koon-Ho Wong, Gary Carter, Ed Dawson, *An analysis of the RC4 family of stream ciphers against algebraic attacks*, in AISC 2010 [**87**] (2010), 67–74. URL: https://crpit.scem.westernsydney.edu.au/abstracts/CRPITV105Wong.html. Citations in this document: §1.3.

[359] Amr M. Youssef, Abderrahmane Nitaj, Aboul Ella Hassanien (editors), *Progress in cryptology—AFRICACRYPT 2013, 6th international conference on cryptology in Africa, Cairo, Egypt, June 22–24, 2013, proceedings*, Springer, 2013. ISBN 978-3-642-38552-0. URL: https://doi.org/10.1007/978-3-642-38553-7. See [246].

[360] Moti Yung (editor), *Advances in cryptology—CRYPTO 2002, 22nd annual international cryptology conference, Santa Barbara, California, USA, August 18–22, 2002, proceedings*, Springer, 2002. ISBN 3-540-44050-X. URL: https://doi.org/10.1007/3-540-45708-9. See [263].

[361] Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, Tal Malkin (editors), *9th international conference on theory and practice in public-key cryptography, New York, NY, USA, April 24–26, 2006, proceedings*, Lecture Notes in Computer Science, 3958, Springer, 2006. ISBN 978–3–540–33851–2. See [42].

A The NISTPQC evaluation criteria

The NISTPQC procedures are mandated by a United States Federal Register notice [**277**] pointing to specific "evaluation criteria which will be used to assess the submissions"; those criteria are listed in [**279**]. This appendix reviews those criteria (minus footnotes), and compares NTRU Prime to the other lattice KEMs according to these criteria.

The "categories" inside the criteria are marked as a *proposal* and are thus not mandated. See Appendix B for analysis of these "categories".

A.1. Security. The following quotes are from [**279**, Section "4.A Security"].

> The security provided by a cryptographic scheme is the most important factor in the evaluation.

The lattice KEMs are all designed to be secure, but the approaches to security are different. The main point of this document is that NTRU Prime's approach minimizes security risks. See especially Sections 1 through 4.

> Schemes will be judged on the following factors:
>
> **4.A.1 Applications of Public-Key Cryptography** NIST intends to standardize post-quantum alternatives to its existing standards for digital signatures (FIPS 186) and key establishment (SP 800-56A, SP 800-56B). These standards are used in a wide variety of Internet protocols, such as TLS, SSH, IKE, IPsec, and DNSSEC. Schemes will be evaluated by the security they provide in these applications, and in additional applications that may be brought up by NIST or the public during the evaluation process. Claimed applications will be evaluated for their practical importance if this evaluation is necessary for deciding which algorithms to standardize.

All of the lattice KEMs are designed to provide IND-CCA2 security. Many protocols can be built from an IND-CCA2 KEM. Prototype software libraries such as Open Quantum Safe [**340**] show how to integrate these KEMs into various applications; these integrations do not ask the KEMs for anything beyond IND-CCA2. The critical question is whether the KEMs *do* provide IND-CCA2 security.

Some KEMs provide further features that *might* have security value. As an example, recall that Kyber [**26**, video, 1:23:00, text on slide] advertised "shared key depends on full transcript" as one of four ways that Kyber is "conservative"; but protocols such as TLS already handle their own transcript hashing, covering more data than a KEM can see. Transcript-hashing security analyses need to be carried out at the protocol layer in any case; see, e.g., [**74**]. The NTRU Prime submission is like Kyber in hashing the full KEM transcript, but warns reviewers "that the security consequences of this hashing need to be formalized and proven, and that it is safer for protocols to rely on simpler promises from primitives". See [**65**, Section 7].

In the context of anonymity, SABER [**34**, Section 7] says that, as a "result of the power-of-two moduli", public keys and ciphertexts are indistinguishable from uniform random strings. It is not correct that this is limited to power-of-2 moduli: standard encoding techniques easily achieve the same for other moduli, as explained in [**65**, Section 4.7, "Encodings of sequences of integers"]. The widely applied pre-quantum baseline here is Elligator [**70**], which encodes points on Montgomery/Edwards curves over prime fields as uniform random strings.

> **4.A.2 Security Definition for Encryption/Key-Establishment** NIST intends to standardize one or more schemes that enable "semantically secure" encryption or key encapsulation with respect to adaptive chosen ciphertext attack, for general use. This property is generally denoted *IND-CCA2 security* in academic literature.

NTRU Prime was very early in recognizing KEMs with IND-CCA2 security as the most important design goal. The original NTRU Prime paper in May 2016 already included a complete reference implementation of a KEM designed for IND-CCA2 security; see [**69**, 2016 version, Figures 2.1–2.2] and [**69**, 2017 version, Appendix Z].

Kyber's predecessor NewHope had been published several months earlier [**16**], but with no protection against chosen-ciphertext attacks. The first version of Frodo [**85**] was published in June 2016, again with no protection against chosen-ciphertext attacks. Adding chosen-ciphertext security is often described as easy but is in fact a major source of security issues, as illustrated by [**64**] breaking HILA5, [**39**] breaking Round2, and various related risks in Table 1.1.

Concrete NTRU proposals before NTRU Prime were typically *PKEs* designed for IND-CCA2 security, more complicated than KEMs designed for IND-CCA2 security. This extra complexity is not justified by typical applications. A KEM transmitting a session key does exactly what is needed for hybrid protocols such as TLS, which use the session key together with symmetric cryptography to transmit many user messages, whereas a PKE transmitting a single user message is aiming at the wrong target.

> The above security definition should be taken as a statement of what NIST will consider to be a relevant attack. Submitted KEM and encryption schemes will be evaluated based on how well they appear to provide this property, when used as specified by the submitter. Submitters are not required to provide a proof of security, although such proofs will be considered if they are available.

Regarding known attacks, the NTRU Prime submission document [**65**, Section 6, pages 46–70] includes a more thorough security evaluation than any of the other lattice-KEM submissions.

Regarding the risk of unknown attacks, NTRU Prime is unique in its approach of proactively eliminating unnecessary attack tools. The security benefits of this approach have already been demonstrated by subsequent attacks within lattice-based cryptography, notably decryption-failure attacks and cyclotomic attacks. See Section 2.

Regarding proofs: For each KEM, there are plausible claims that QROM IND-CCA2 security of the KEM is provably related to a simpler security property of the underlying mathematical PKE. To *some* extent these claims are backed by literature giving careful proofs: see, e.g., [**73**]. Known proofs are

- stronger for `ntruhps`, `ntruhrss`, and `sntrup` than for `ntrulpr` (because of derandomization), and
- stronger for `ntrulpr` than for `saber`, `kyber`, and `frodo` (because of the rate of decryption failures).

Most of the attack advances in Section 1 and risks in Table 1.1 are regarding the mathematical PKEs. Available proofs did not stop these attacks and do not control these risks. See [**49**] and Section 5.

> For the purpose of estimating security strengths, it may be assumed that the attacker has access to the decryptions of no more than 2^{64} chosen ciphertexts; however, attacks involving more ciphertexts may also be considered. Additionally, it should be noted that NIST is primarily concerned with attacks that use classical (rather than quantum) queries to the decryption oracle or other private-key functionality.

User devices are of course limited in how much computation they will carry out. The exact limit does not seem relevant to comparisons of these KEMs.

> **4.A.3 Security Definition for Ephemeral-Only Encryption/Key-Establishment** While chosen ciphertext security is necessary for many existing applications (for example, nominally ephemeral key exchange protocols that allow key caching), it is possible to implement a purely ephemeral key exchange protocol in such a way that only passive security is required from the encryption or KEM primitive.
>
> For these applications, NIST will consider standardizing an encryption or KEM scheme which provides semantic security with respect to chosen plaintext attack. This property is generally denoted *IND-CPA security* in academic literature.
>
> The above security definition should be taken as a statement of what NIST will consider to be a relevant attack. Submitted KEM and encryption schemes will be evaluated based on how well they appear to provide this property, when used as specified by the submitter. Submitters are not required to provide a proof of security, although such proofs will be considered if they are available. Any security vulnerabilities that result from re-using a key should be fully explained.

All of these submissions spend some cycles on protection against chosen-ciphertext attacks. Some submissions advertise the potential speedup of using a key just once, but these speedups are (1) small in context and (2) outweighed by the cost of generating a new key for every ciphertext; the benefit is thus limited to a small speedup for applications that have other reasons to generate a new key for every ciphertext. Meanwhile supporting any such option creates risks, as stated in [**226**]: "CPA vs CCA security is a subtle and dangerous distinction, and if we're going to invest in a post-quantum primitive, better it not be fragile." Systems-security issues are not generally covered by the evaluation criteria, but NTRU Prime's focus on an IND-CCA2 KEM as the sole user interface should at least be recognized under the simplicity criterion below.

Regarding "purely ephemeral", the intent appears to have been to refer to *one-time* keys. It is important to realize that (1) "one-time" is neither necessary nor sufficient for "ephemeral"; (2) "ephemeral" is the right concept for the security goal at hand; and (3) this affects performance evaluations. See Section 6.3.

> **4.A.4 Security Definition for Digital Signatures** [not relevant here; omitted]

4.A.5 Security Strength Categories [covered in Appendix B]

4.A.6 Additional Security Properties While the previously listed security definitions cover many of the attack scenarios that will be used in the evaluation of the submitted algorithms, there are several other properties that would be desirable:

> One such property is perfect forward secrecy. While this property can be obtained through the use of standard encryption and signature functionalities, the cost of doing so may be prohibitive in some cases. In particular, public-key encryption schemes with a slow key generation algorithm, such as RSA, are typically considered unsuitable for perfect forward secrecy. This is a case where there is significant interaction between the cost, and the practical security, of an algorithm.

The SUPERCOP benchmarking framework reports RSA-2048 key generation (using OpenSSL 1.1.1) taking a median of 257 million cycles on a Haswell core (`hiphop`, `supercop-20210604`). For comparison, [66] reports just 156317 cycles per key for `sntrup761`: 1600 times faster, with a much higher security level against known attacks. The overall CPU cycles used per TLS 1.3 handshake in [66], including a new `sntrup761` key for each handshake, are smaller than the cycles used on the same CPU for X25519, the standard choice of curve for TLS 1.3; a hybrid of both cryptosystems is easily affordable and is less expensive than NIST P-256, the other standard option in TLS 1.3. See also Sections 6.3 and 7.14. NTRU Prime also has an `ntrulpr` option with even faster key generation, but this option is overkill for purposes of this evaluation criterion.

Available NTRU software does not yet include keygen speedups analogous to [66], but it would be unreasonable to claim on this basis that NTRU does not support "perfect forward secrecy". If there is a performance problem with new keys for any of these KEMs, the problem is much more likely to come from communication cost than from computation time. See Section 6.6.

> Another case where security and performance interact is resistance to side-channel attacks. Schemes that can be made resistant to side-channel attack at minimal cost are more desirable than those whose performance is severely hampered by any attempt to resist side-channel attacks. We further note that optimized implementations that address side-channel attacks (e.g., constant-time implementations) are more meaningful than those which do not.

The NTRU Prime C software, both reference and optimized, has always been constant-time. As mentioned in Section 3.6, the software is written within a framework that verifies immunity to timing attacks. On the other hand, any KEMs not in this state should be expected to catch up eventually.

As for more invasive side channels: SABER was early in producing a masked implementation, but that implementation was broken by [280]. Protection "at minimal cost" does not seem to be achievable for any of these KEMs; see [28].

The path towards NTRU Prime implementations with advanced side-channel protection is described in Section 7.10.

In principle, masking of any order can be applied to any computation, but the costs depend on the computation. For example, linear operations, such as multiplication by a public ring element, are generally less expensive to mask than non-linear operations, such as hashing. It is thus not safe to predict costs of masked implementations given only costs of unmasked implementations.

The SABER submission suggests [**34**, Section 7] that prime moduli are a speed problem for conversion between arithmetic masking and xor masking. However, these conversions are only a small part of the cost of a masked KEM, whereas NTRU and NTRU Prime should be expected to have larger advantages in bit operations for multiplications; see Section 6.5. The SABER submission also suggests that masked noise generation is a speed problem compared to rounding; NTRU Prime uses rounding too.

In the aforementioned context of anonymity, the SABER submission says "Saber is naturally constant time over different public keys in contrast to prime-moduli schemes". This issue applies to the prime moduli in Kyber but does *not* apply to the prime moduli in NTRU Prime. There is a different issue for `kyber90s` and `ntrulpr`, namely that if the symmetric primitive used for LPR generator expansion is chosen to be AES then on many platforms the public key will leak through timing. None of this applies to `sntrup`.

> A third desirable property is resistance to multi-key attacks. Ideally an attacker should not gain an advantage by attacking multiple keys at once, whether the attacker's goal is to compromise a single key pair, or to compromise a large number of keys.

Most cryptanalytic papers focus on single-target attacks. The exceptions have found many speedups for multi-target attacks; see, e.g., the discrete-logarithm speedups listed in [**46**]. So there is clearly a risk here, and the risk has even less evaluation than the risk of single-target attacks.

Some submissions advertise public-key hashing in the context of multi-target protection. (NTRU Prime also hashes the public key.) Public-key hashing can help against *some* types of multi-target attacks. However, the general multi-target attack surface for lattice KEMs is much larger than the corner addressed by public-key hashing.

Fortunately, the multi-target risk has a clear quantitative limit that is small enough to be useful. The NTRU Prime submission [**65**, Section 7.1] says "We recommend handling multi-target attacks by aiming for a very high single-target security level, and then relying on the fact that T-target attacks gain at most a factor T. This approach is simple and effective, and is not much more expensive than merely stopping single-target attacks."

> A final desirable, although ill-defined, property is resistance to misuse. Schemes should ideally not fail catastrophically due to isolated coding errors, random number generator malfunctions, nonce reuse, keypair reuse (for ephemeral-only encryption/key establishment) etc.

See above regarding implementation security.

> **4.A.7 Other Consideration Factors** As public-key cryptography tends to contain subtle mathematical structure, it is very important that the mathematical structure be well understood in order to have confidence in the security of a cryptosystem. To assess this, NIST will consider a variety of factors. All other things being equal, simple schemes tend to be better understood than complex ones. Likewise, schemes whose design principles can be related to an established body of relevant research tend to be better understood than schemes that are completely new, or schemes that were designed by repeatedly patching older schemes that were shown vulnerable to cryptanalysis.

All of the lattice KEMs score poorly on this criterion. If "well understood" is a yes-no prerequisite for standardization then none of these KEMs should be standardized. See Section 1.

A closer look shows that some of the KEMs are structurally immune to *some* classes of lattice attacks. See Table 1.1. In particular, NTRU Prime's `ntrulpr` structurally avoids three risks incurred by `saber` and `kyber` without incurring any additional risks, and NTRU Prime's `sntrup` structurally avoids two risks incurred by `ntruhrss` and `ntruhps` without incurring any additional risks.

The comparison with Frodo is more difficult, since Frodo incurs some risks while avoiding others. See Section 4.

Regarding "simple schemes", the original NTRU Prime reference software [**69**, Appendix Z] fits into two pages, and all of the subsequent lattice KEMs under consideration have similarly concise descriptions. However, it is important to realize that the *security analysis* is vastly more complicated. See the many papers listed in Section 1. This is not a new phenomenon in cryptography: consider, e.g., RC4, which is a very simple scheme but has a very complicated security analysis.

> NIST will also consider the clarity of the documentation of the scheme and the quality of the analysis provided by the submitter. Clear and thorough analysis will help to develop the quality and maturity of analysis by the wider community. NIST will also consider any security arguments or proofs provided by the submitter. While security proofs are generally based on unproven assumptions, they can often rule out common classes of attacks or relate the security of a new scheme to an older and better studied computational problem.
>
> In addition to NIST's own expectations for the scheme's long-term security, NIST will also consider the judgment and opinions of the broader cryptographic community.

See above regarding proofs and quality of analysis. Regarding maturity of analysis, NTRU Prime is the oldest of the lattice KEMs under consideration; see the "instability" rows in Table 1.1.

A.2. Cost. The following quotes are from [279, Section "4.B Cost"].

> As the cost of a public-key cryptosystem can be measured on many different dimensions, NIST will continually seek public input regarding which performance metrics and which applications are most important. If there are important applications that require radically different performance tradeoffs, NIST may need to standardize more than one algorithm to meet these diverse needs.

The available evidence indicates that communication costs are dominant for these lattice KEMs; see Section 6.6. All of the lattice KEMs except for Frodo have similar communication costs; see Figure 7.3.

Other metrics show larger performance differences, depending on the metric and the target security level. See, e.g., Figures 7.5 and 7.6. However, it seems difficult to point to diverse application performance requirements as an argument for standardizing more than one of NTRU, NTRU Prime, SABER, and Kyber.

> **4.B.1 Public Key, Ciphertext, and Signature Size** Schemes will be evaluated based on the sizes of the public keys, ciphertexts, and signatures that they produce. All of these may be important consideration factors for bandwidth-constrained applications or in Internet protocols that have a limited packet size. The importance of public-key size may vary depending on the application; if applications can cache public keys, or otherwise avoid transmitting them frequently, the size of the public key may be of lesser importance. In contrast, applications that seek to obtain perfect forward secrecy by transmitting a new public key at the beginning of every session are likely to benefit greatly from algorithms that use relatively small public keys.

NTRU Prime scores well here—as do NTRU, SABER, and Kyber. The winner depends on the target security level. See Figure 7.3.

In "bandwidth-constrained applications or in Internet protocols that have a limited packet size", the highest security could be from NTRU Prime, or from NTRU, or from SABER (in some cases matched by Kyber), depending on the exact size constraint. Again, see Figure 7.3.

Each of these KEMs has public-key size close to ciphertext size, but there are slight differences, as illustrated by `sntrup` (smaller ciphertexts) and `ntrulpr` (smaller keys), so it is helpful to establish the role of these sizes:

- In applications that transmit a new key for every ciphertext, the correct metric is pk+ct.
- In applications where keys are cached, the correct metric is ct plus a fraction of pk that depends on the cache effectiveness. Even if the cache works for only 90% of the key lookups, the effective fraction of pk drops to 10%, and for these KEMs the resulting metric is tantamount to simply ct.

Given the $\approx 2\times$ performance improvement from caching, the first step towards improving performance in performance-sensitive applications is to cache keys, so the ct scenario should be given higher weight than the pk+ct scenario in performance evaluations. See [258] and Section 3.1. Contrary to popular belief, "forward secrecy" is fully compatible with the first scenario; see Section 6.3. Regarding TLS, see Section 7.16.

> **4.B.2 Computational Efficiency of Public and Private Key Operations** Schemes will also be evaluated based on the computational efficiency of the public key (encryption, encapsulation, and signature verification) and private key (decryption, decapsulation, and signing) operations. The computational cost of these operations will be evaluated both in hardware and software. The computational cost of both public and private key operations is likely to be important for almost all operations, but some applications may be more sensitive to one or the other. For example, signing or decryption operations may be done by a computationally constrained device like a smartcard; or alternatively, a server dealing with a high volume of traffic may need to spend a significant fraction of its computational resources verifying client signatures.

NTRU Prime scores well here too, and is again often the winner, depending on the application environment. See, e.g., Figures 7.5 and 7.6 for encapsulation and decapsulation operations on "a computationally constrained device like a smartcard", specifically on an ARM Cortex-M4 microcontroller, and Figure 7.9 for secret-key size.

Regarding "evaluated both in hardware and software", it is important to realize that almost all NISTPQC speed evaluations have been in environments with fast multipliers. This includes Cortex-M4 evaluations and FPGA evaluations. True hardware evaluations—ASIC evaluations—see the intrinsic cost of each multiplication; as noted earlier, this is favorable to NTRU and NTRU Prime, less favorable to SABER, and unfavorable to Kyber. See Section 6.5.

> **4.B.3 Computational Efficiency of Key Generation** Schemes will also be evaluated based on the computational efficiency of their key generation operations, where applicable. As noted in Section 4.A.6, the most common scenario where key generation time is important is when a public-key encryption algorithm or a KEM is used to provide perfect forward secrecy. Nonetheless, it is possible that key generation times may also be important for digital signature schemes in some applications.

See above regarding key generation.

> **4.B.4 Decryption Failures** Some public-key encryption algorithms and KEMs, even when correctly implemented, will occasionally produce ciphertexts that cannot be decrypted/decapsulated. For most applications, it is important that such decryption failures be rare or absent. For algorithms with decryption/decapsulation failures, submitters must provide the failure rate, as well as an analysis of the impact on security that these failures could cause. While applications can always obtain an

> acceptably low decryption failure rate by encrypting the same plaintext multiple times, and interactive protocols can simply restart when key establishment fails, these types of solutions have their own performance costs.

The view of decryption failures as a *performance* issue is not relevant to any of these KEMs: decryption failures are too rare to be triggered by normal usage. However, there is a *security* risk: *attack* algorithms can search many more ciphertexts, and perhaps can recognize failing ciphertexts. See Section 3.9 and Section 5.7. From this security perspective, NTRU and NTRU Prime have an advantage over Kyber, SABER, and Frodo: NTRU and NTRU Prime are perfectly correct, i.e., have no decryption failures.

A.3. Flexibility, simplicity, adoption. The following quotes are from [**279**, Section "4.C Algorithm and Implementation Characteristics"].

> **4.C.1 Flexibility** Assuming good overall security and performance, schemes with greater flexibility will meet the needs of more users than less flexible schemes, and therefore, are preferable.
>
> Some examples of "flexibility" may include (but are not limited to) the following:
>
> a. The scheme can be modified to provide additional functionalities that extend beyond the minimum requirements of public-key encryption, KEM, or digital signature (e.g., asynchronous or implicitly authenticated key exchange, etc.).

See above regarding IND-CCA2 etc. More advanced protocols such as fully homomorphic encryption cannot be built from these KEMs; see Section 1.4.

> b. It is straightforward to customize the scheme's parameters to meet a range of security targets and performance goals.

This is a clear win for NTRU Prime compared to SABER and Kyber. See below regarding NTRU.

Structurally, SABER and Kyber support only dimensions that are multiples of 256, while NTRU Prime supports many intermediate dimensions. See Section 6.7; also compare the large `kyber` jumps in, e.g., Figure 7.6 to the demonstrated flexibility of `sntrup` (red dots).

Many more NTRU Prime parameter sets, including intermediate dimensions and larger dimensions, have their Core-SVP levels and sizes displayed in [**65**, pages 104–111]. The generation and Core-SVP evaluation of parameter sets are fully automated.

As noted in Section 4, larger parameter sets appear problematic for Kyber. There are two basic reasons for this:

- The Kyber modulus 3329 causes increasing decryption-failure difficulties as the dimension increases. It is not clear how to compensate for this without (1) significantly reducing Core-SVP or (2) revisiting Kyber's structural decision to use the same modulus for all parameter sets.

- The matrix structure of Kyber produces quadratic slowdowns. For example, merely *generating* the matrix reportedly takes 318000 Cortex-M4 cycles for `kyber768` and 713000 Cortex-M4 cycles for `kyber1024`, and each operation also has to *use* the entire matrix.

Interestingly, even though Kyber's limitation to $(\mathbf{Z}/3329)[x]/(x^{256}+1)$ creates these difficulties and directly damages Kyber's performance in, e.g., Figure 7.5, Kyber advertises this as a performance feature: "Optimized implementations only have to focus on a fast dimension-256 NTT and a fast Keccak permutation. This will give very competitive performance *for all parameter sets of* Kyber." This is listed in [**25**, Section 7] as the first "unique advantage" of Kyber. This advantage is summarized as "Ease of implementation", but the way that NTTs are baked into the Kyber specification makes *simple* implementations of Kyber more complicated than simple implementations of competitors; evidently the intent was to say how easy *optimized* implementations are.

For comparison, the AVX2-optimized software released for NTRU Prime for all parameter sets is automatically generated from a unified code base, sharing a size-512 NTT across all sizes. The reference software is also shared across sizes. The same generator trivially produces optimized code for intermediate dimensions that Kyber, because of its module structure, is structurally incapable of handling. The Kyber submission is incorrect when it claims [**25**, Section 6.4.3] that changing ring requires "completely re-implementing all the operations".

It is easy to see from the NTRU specification that NTRU can be customized to meet a range of security targets and performance goals, the same way that NTRU Prime can. The reason this is less obvious for NTRU than for NTRU Prime in Figure 7.5 is that NTRU has specified only four `ntruhps` sizes and two `ntruhrss` sizes, and has released optimized code for fewer sizes. The NTRU *documentation* includes Core-SVP evaluations of many more parameter sets.

> c. The algorithms can be implemented securely and efficiently on a wide variety of platforms, including constrained environments, such as smart cards.

All of these KEMs can be implemented on a wide range of platforms, except that Frodo would have trouble on *extremely* small devices. Efficiency varies: see Section 7.4 for NTRU Prime's performance wins on an ARM Cortex-M4 microcontroller. Kyber is unusual in having its performance profile tilted towards large devices; see Section 6.5.

Regarding secure implementation, important challenges include protecting against side channels (see above) and eliminating bugs. Some submissions point to specific implementation features (e.g., power-of-2 moduli in Table 1.1 mean that software does not have to reduce modulo odd primes) and claim that these features are important advantages, but these claims are negatively correlated with observed implementation security. For example, the Frodo submission [**15**, Section 6.1, "Ease of implementation"] stated as Frodo's first "advantage" that

"One of the features of FrodoKEM is that it is easy to implement and naturally facilitates writing implementations that are compact and run in constant-time", but Frodo's official software was then

- broken by the timing attack of [**169**],
- modified to avoid the timing attacks, and then
- broken by an even easier attack [**321**] because of a bug in the modification.

The NTRU Prime submission is at the opposite extreme. The NTRU Prime web pages have always included a dedicated "Warnings" page [**284**] covering, among other things, the need for software review. The software has, since 2019, been factored into modules with separate tests and optimizations. As mentioned in Section 3.6, new tools [**61**] have computer-verified for most of those modules that

- the existing AVX2-optimized implementation and
- the existing (simpler) reference implementation

produce the same outputs for all possible inputs. Some modules, notably the multiplier, are too large for the current tools to handle, but [**66**, Appendix A] announced computerized range-checking of the critical NTT software inside the AVX2-optimized multiplier. Work is continuing towards verification of all subroutines and towards verification for more platforms.

> d. Implementations of the algorithms can be parallelized to achieve higher performance.

All of the lattice KEMs have demonstrated successful vectorization. Further parallelization is clearly possible for the multiplications. Parallel hash functions are also available, although care is required regarding security; see Sections 3.15 and 3.16.

> e. The scheme can be incorporated into existing protocols and applications, requiring as few changes as possible.

It isn't clear what this means for KEMs beyond the performance requirements (e.g., ciphertext size). Aside from performance, all of the KEMs have the same externally visible data flow.

> **4.C.2 Simplicity** The submitted scheme will be judged according to its relative design simplicity.

The KEM designs have many similarities, but each KEM (even if claimed to be "simple") has complications that do not appear in some of the other KEMs. For example:

- The underlying PKEs are simpler for Quotient NTRU than for Product NTRU: for example, the Quotient NTRU ciphertext is just $Gb + d$, whereas the Product NTRU ciphertext is $(Gb + d, Ab + M + c)$.

- Product NTRU needs extra work for derandomization, while Quotient NTRU does not.
- Combining polynomials with matrices (modules, as in Kyber and SABER) is more complicated than using just polynomials (NTRU, NTRU Prime) or using just matrices (Frodo).
- Using binomials $x^n + 1$ (Kyber, SABER) is simpler than using $x^p - x - 1$ (NTRU Prime), which in turn is simpler than using $x^p - 1$ with its factor $x - 1$ (NTRU).
- Kyber's NTT-based representation of polynomials is more complicated than the traditional representation used in NTRU, NTRU Prime, and SABER.
- The error distribution in Frodo is more complicated than the sum-of-bits error distributions in Kyber and SABER and the ternary distributions in NTRU and NTRU Prime.

The best way to see what varies from one KEM to another is through detailed cross-KEM comparison charts, as in [**49**, pages 48–52] and Table 1.1. Another source of information regarding Quotient NTRU vs. Product NTRU is the NTRU Prime software, which is mostly shared but has, e.g., inversion code only for Quotient NTRU and derandomization code only for Product NTRU.

> **4.C.3 Adoption** Factors that might hinder or promote widespread adoption of an algorithm or implementation will be considered in the evaluation process, including, but not limited to, intellectual property covering an algorithm or implementation and the availability and terms of licenses to interested parties. NIST will consider assurances made in the statements by the submitter(s) and any patent owner(s), with a strong preference for submissions as to which there are commitments to license, without compensation, under reasonable terms and conditions that are demonstrably free of unfair discrimination.

This is a problem for `ntrulpr`, `saber`, and `kyber`, and appears to be decisive regarding those KEMs since the call for submissions says that it is "critical that this process leads to cryptographic standards that can be freely implemented in security technologies and products". See Sections 3.17 and 4.3.

OpenSSH integrated the round-1 version of `sntrup761` as a fully supported SSH key-exchange option in April 2019, and replaced it with the current version of `sntrup761` (which has the same underlying mathematical PKE) in March 2021 [**291**]. OpenBSD now [**290**] supports `sntrup761` for IPsec key exchange. Google and Cloudflare have carried out web-browsing experiments [**217**] with a variant of `ntruhrss701`.

B The NISTPQC categories

The standard scientific way to understand tradeoffs between two variables—for example, performance vs. security—is with two-dimensional scatterplots, such as Figures 7.3, 7.5, 7.6, and 7.9. There are some common pitfalls in the graphing process, but these pitfalls are avoidable. See [50, Section 3].

The scatterplots are only as good as the data they use. For the figures above, the numbers on the vertical axis are Core-SVP; recall from Section 3.4 that Core-SVP is a combination of underestimates, overestimates, possible underestimates, and possible overestimates of the number of operations used in certain lattice attacks. The standard scientific reaction to crude models is to study the topic more closely and build better models, as in [65, Section 6].

Comparing costs of attacks with different mixes of operations—for example, asking whether a lattice attack is as cheap as an attack against AES-128—generally requires defining a cost metric, a way to compute a cost for each operation. The details are important. There are many examples in the literature of algorithms where different choices of cost metrics make a huge difference in algorithm cost, often reversing comparisons between algorithms and comparisons between problems. See, e.g., [91] and [211, Section 5.4].

With these issues and standard scientific practices in mind, let's look at what NIST tried to accomplish with its "categories", and what actually happened.

B.1. Symmetric primitives as a security floor. NIST's August 2016 draft of the evaluation criteria [278, Section 4.A.4, "Target security strengths"] began by asking submitters "to provide parameter sets that meet or exceed each of five target security strengths". Submitters were asked to be "confident that the specified security target is met or exceeded".

"Strength" 1 was labeled as "128 bits classical security / 64 bits quantum security", the stated intent being to be at least as secure as "brute-force attacks against AES-128". "Strength" 2 was labeled as "128 bits classical security / 80 bits quantum security", the stated intent being to be at least as secure as "brute-force collision attacks against SHA-256/ SHA3-256". Et cetera.

This approach was critiqued in [46]:

> Quantitatively comparing post-quantum public-key security levels is going to be a nightmare. I see only two ways that submitters a year from now can possibly be "confident that the specified security target is met or exceeded": (1) overkill; (2) overconfidence. Many users will not be satisfied with overkill, and NIST should not encourage overconfidence.

Various illustrative examples and (correct) predictions of evaluation problems appeared in [46], along with a proposal to focus on the problem at hand ("Scrap the requirement of a pre-quantum security analysis. Users will use cheap ECC hybrids to obtain the pre-quantum security that they want") and to prioritize accuracy ("Ask them to do the most accurate job that they can of analyzing post-quantum security. Don't ask for fake confidence").

B.2. Pseudo-definitions of "category" boundaries. NIST's final call for submissions [279] in December 2016 renamed "strengths" as "categories", said that each "category" will be defined by a symmetric primitive "whose security will serve as a floor", and made some changes in the details—including a giant leap of security for SHA3-256, which was assigned

- “128 bits classical security / 80 bits quantum security” in [278], but
- “2^{146} classical gates” with no quantum speedup in [279].

The leap from 2^{80} to 2^{146} is much larger than can be explained by a switch from counting hash calls to counting “gates”.

The Brassard–Høyer–Tapp algorithm [90] finds SHA3-256 collisions in about 2^{83} quantum operations. These operations use far fewer than 2^{146} quantum “gates” with the set of “gates” defined by, e.g., Ambainis [17]. Clearly NIST must have in mind a more restrictive set of “gates”—but which set? A definition is critical for comparisons. Five years later, despite clarification requests, NIST's “categories” still do not have clear definitions. See [53, Section 5.4] for further analysis.

B.3. Lattice guesswork. Analyzing the number of operations used in known lattice attacks for cryptographic sizes remains an open research problem—see [65, Section 6]—even if faster attacks are assumed not to exist. Lattice submissions in round 1 of NISTPQC took different approaches to handling (1) the unknown number of operations used in these algorithms and (2) the lack of clarity from NIST regarding how to assign operation counts to “categories”.

Let's look in particular at Kyber. Like NewHope, Kyber calculated “the core-SVP hardness” of its parameter sets, and said [24, page 17] that this is “a very conservative lower bound on the cost of an actual attack”. However, this “lower bound” was far below 2^{128} for `kyber512`. So the submission argued that, because of various other unquantified factors, `kyber512` was still as difficult to break as AES-128, meaning that `kyber512` qualified for “category” 1.

But is it true that `kyber512` qualified for “category” 1? Does the round-3 `kyber512`, a patched version of the previous `kyber512`, qualify for “category” 1? The answers to these questions remain unclear even if, again, faster attacks are assumed not to exist. See Section 1.3.

B.4. How “categories” have damaged analyses. See [53] for examples of how NIST's emphasis on “categories” has damaged performance evaluations. What follows is an example of how the same emphasis has damaged security evaluations.

Consider the problem of solving dimension-β SVP. The best enumeration algorithms known have exponent $\Theta(\beta \log \beta)$. The best sieving algorithms known have exponent $\Theta(\beta)$, and are therefore faster for all sufficiently large β.

However, this does not say anything about concrete sizes. It is important to realize that the slow growth of $\log \beta$ makes the cutoff between sieving and enumeration extremely sensitive to improvements in either algorithm. See the numerical examples in [65, Section 6.9].

When NISTPQC began, enumeration had asymptotic exponent approximately $\approx 0.187\beta \log_2 \beta$, and sieving had asymptotic exponent $\approx 0.292\beta$. Including lower-order terms is more favorable to enumeration than to sieving, and quantum computers chop the enumeration exponent in half (see [18]) while reducing the sieving exponent by only about 10%, but all this is outweighed by $\log_2 \beta$ once β is large enough. The question is how large.

The round-1 Kyber submission stated [24, Section 5.1.1] that "sophisticated enumeration, with serious optimization … and with quantum speedups", was outperformed by sieving "for dimensions larger than 250, quite possibly already earlier", and was thus not a threat since the "smallest dimension that we are interested in for the cryptanalysis of Kyber is 390". The sieving-vs.-enumeration comparison was under an assumption "that access into even exponentially large memory is free"; this is important because sieving is memory-intensive while enumeration is not.

In other words, the submission was saying that sieving-with-free-memory was cheaper than enumeration. Obviously it's also cheaper than sieving-with-real-memory, so an attack-cost analysis based on sieving-with-free-memory would be a lower bound for an attack-cost analysis based on enumeration or sieving-with-real-memory.

Then attacks advanced. Section 1 noted dramatic recent improvements in enumeration speed, including [9], which reduced the asymptotic exponent from $\approx 0.187\beta\log_2\beta$ to $\approx 0.125\beta\log_2\beta$, and [10, page 547], which estimated the new cutoff between quantum enumeration and quantum sieving as $\beta = 547$, far above the 250 mentioned above. Every proposed lattice system with pre-quantum Core-SVP below about 2^{160} has, as a direct result of these attacks, less post-quantum security than previously believed.

Seeing exponents drop by $1.5\times$, and seeing various post-quantum proposals suddenly having less post-quantum security, perfectly illustrates the instability of security analysis of lattice-based cryptography. There is a clear risk of further enumeration improvements further damaging post-quantum security and even damaging pre-quantum security. The advance in enumeration also means that the known pre-quantum-to-post-quantum loss for lattice systems is larger than it was before. But now let's look at what "categories" do to this analysis.

AES has an even larger pre-quantum-to-post-quantum loss. The question of *whether a lattice KEM reaches the AES floor* is thus unable to see the known loss of post-quantum security. As a direct result of NIST's promotion of "categories", this question is emphasized in [10, Section 1], which says that "this work does *not* invalidate the claimed NIST Security Level" of the affected systems.

The round-3 Kyber submission claims that round-3 `kyber512` qualifies for "category" 1, as hard to break as AES-128. The analysis in the submission says that a sieving attack on round-3 `kyber512` could use as few as $2^{135.5}$ "gates"—which, according to the submission, is possible for known attacks—would not be "catastrophic, in particular given the massive memory requirements that are ignored in the gate-count metric". Let's review the logic here:

- The round-3 submission says a sieving attack against round-3 `kyber512` could use hundreds of times fewer "gates" than an AES-128 attack.
- How, then, does round-3 `kyber512` qualify for "category" 1? Answer: this potential speedup over AES-128 is sieving-with-free-memory; sieving-with-real-memory is much more expensive. This answer sounds reasonable if the cost metric includes realistic costs for memory.

- But what about enumeration? Answer: enumeration doesn't matter, since for these sizes it seems more expensive than sieving-with-free-memory. This answer sounds reasonable if the cost metric includes free memory—but then how can `kyber512` qualify for "category" 1?

NIST's "categories" are being interpreted as allowing free memory for purposes of skipping an enumeration analysis, but as including realistic costs for memory for purposes of allowing `kyber512` to qualify for "category" 1. These inconsistent interpretations are enabled by NIST's continued failure to define a cost metric.

A back-of-the-envelope calculation suggests that—assuming realistic memory costs, as `kyber512` does in its argument to qualify for "category 1"—the latest enumeration algorithms are

- easily the fastest known quantum attacks against `kyber512` and
- quite possibly the fastest known non-quantum attacks against `kyber512`—more research is needed regarding lower-order factors.

Even if the algorithms are still slower than AES-128 key search, this change in the security picture is big news. Focusing on "categories" suppresses this news. Scientific quantification is buried under the question of whether a KEM reaches a particular "floor".

B.5. What "categories" were supposed to accomplish. Recall that the "Target security strengths" section of NIST's August 2016 draft [**278**] started by asking submitters "to provide parameter sets that meet or exceed each of five target security strengths". NIST's goal was clear: to be able to say that each of the symmetric primitives was matched or surpassed in strength by a post-quantum parameter set.

In the final call [**279**], this no longer sounded like the goal of the "categories". Instead "categories" were portrayed as addressing other problems. Let's look at this list of problems and at how "categories" have failed to help.

> NIST anticipates that there will be significant uncertainties in estimating the security strengths of these post-quantum cryptosystems. These uncertainties come from two sources: first, the possibility that new quantum algorithms will be discovered, leading to new cryptanalytic attacks; and second, our limited ability to predict the performance characteristics of future quantum computers, such as their cost, speed and memory size.

This is a surprisingly limited list of sources of uncertainty regarding security. What about advances in *non-quantum* algorithms? What about difficulties in analyzing the *number of operations* used in algorithms, even when the cost of each operation is well understood?

The normal scientific way to handle uncertainties regarding the performance of future quantum computers is to analyze multiple possibilities. For example, [**47**] reviewed the difficulty of carrying out 2^{128} operations and then compared the following possibilities:

- "quantum op costs 2^{10} pre-q ops $\Rightarrow$ 2^{118} quantum ops aren't a threat"

- "quantum op costs 2^{20} pre-q ops $\Rightarrow$ 2^{108} quantum ops aren't a threat"
- "quantum op costs 2^{30} pre-q ops $\Rightarrow$ 2^{98} quantum ops aren't a threat"
- "quantum op costs 2^{40} pre-q ops $\Rightarrow$ 2^{88} quantum ops aren't a threat"
- "quantum op costs 2^{50} pre-q ops $\Rightarrow$ 2^{78} quantum ops aren't a threat"

Different possibilities could end up favoring different submissions: for example, aiming for more attack operations is less of an issue for a submission with linear scalability than for a submission with quadratic scalability.

As for the possibility of unknown attacks, cryptographers traditionally handle this possibility by searching for attacks—and, sometimes, by explicitly managing risks of attack advances, as in NTRU Prime.

> In order to address these uncertainties, NIST proposes the following approach. Instead of defining the strength of a submitted algorithm using precise estimates of the number of "bits of security," NIST will define a collection of broad security strength categories. Each category will be defined by a comparatively easy-to-analyze reference primitive, whose security will serve as a floor for a wide variety of metrics that NIST deems potentially relevant to practical security.

NIST's pseudo-definitions of low-precision "categories" haven't done anything to address uncertainties regarding new quantum algorithms and the performance of quantum computers. On the contrary, by emphasizing comparisons to AES and thus emphasizing pre-quantum security, these "categories" took attention away from what should have been the whole point of NISTPQC: protecting against quantum computers. This was predictable, and was predicted in [**46**].

> A given cryptosystem may be instantiated using different parameter sets in order to fit into different categories. The goals of this classification are:
>
> 1. To facilitate meaningful performance comparisons between the submitted algorithms, by ensuring, insofar as possible, that the parameter sets being compared provide comparable security.

Standard scientific scatterplots, such as Figure 7.3, directly show security-performance tradeoffs (1) within each submission and (2) across submissions. Low-precision "categories" are a giant step backwards in comparisons, hiding most of this information. Compare [**53**, Figures 4.2 and 4.5] for a visualization of how much information is lost.

> 2. To allow NIST to make prudent future decisions regarding when to transition to longer keys.

Everyone who has an idea of what "category 1" means also knows how to compare a number to 128.

> 3. To help submitters make consistent and sensible choices regarding what symmetric primitives to use in padding mechanisms or other components of their schemes requiring symmetric cryptography.

Most submissions, including NTRU Prime, sensibly choose to use ≥256-bit symmetric cryptography everywhere, even when the target security level is lower. This simplifies the review process and has negligible impact on performance. A few submissions, such as Frodo, use smaller symmetric cryptography to be "consistent" with the target security level.

> 4. To better understand the security/performance tradeoffs involved in a given design approach.

This differs from #1 above: #1 was about comparisons between designs, whereas #4 is about tradeoffs inside one design. "Categories" don't completely hide the jumps in Kyber cost—where's the "category 2" Kyber parameter set?—but scientific scatterplots such as Figure 7.5 do much better at showing the full picture.

> In accordance with the second and third goals above, NIST will base its classification on the range of security strengths offered by the existing NIST standards in symmetric cryptography, which NIST expects to offer significant resistance to quantum cryptanalysis. In particular, NIST will define a separate category for each of the following security requirements (listed in order of increasing strength):
>
> [breaking categories 1, 2, 3, 4, 5 "must require computational resources comparable to or greater than those required for" AES-128 key search, SHA-256 collision search, AES-192 key search, SHA-384 collision search, AES-256 key search; resource-estimation details omitted]
>
> NIST asks submitters to provide a preliminary classification, according to the above categories, for all parameter sets that they intend to be considered for standardization. All submitters are advised to be somewhat conservative in their preliminary classifications, but submitters of algorithms where the complexity of the best known attack has recently decreased significantly, or is otherwise poorly understood, should be especially conservative.

Here NIST's goal is again clear: to be able to say that each of the symmetric primitives is matched or surpassed in strength by a post-quantum parameter set. This is, however, a distraction from the scientific task of evaluating attacks.

> NIST will not require submitters to provide distinct parameter sets for all five security strength categories. Submitted parameter sets meeting the requirements of a higher category will be automatically considered to meet the requirements of all lower categories. Submitters may also provide more than one parameter set in the same category, in order to demonstrate how parameters can be tuned to offer better performance or higher security margins.

The reference to "higher security margins" within "the same category" suggests that "categories" are not the end of the story. However, NIST's text puts much more emphasis on “categories”, and these “categories” have played a major role in NISTPQC. Scatterplots do a much better job of showing the same information.

> NIST recommends that submitters primarily focus on parameters meeting the requirements for categories 1, 2 and/or 3, since these are likely to provide sufficient security for the foreseeable future. To hedge against future breakthroughs in cryptanalysis or computing technology, NIST also recommends that submitters provide at least one parameter set that provides a substantially higher level of security, above category 3. Submitters can try to meet the requirements of categories 4 or 5, or they can specify some other level of security that demonstrates the ability of their cryptosystem to scale up beyond category 3.

All of this could have been stated just as easily in terms of quantitative security levels.

B.6. The way forward. What the evaluation criteria *require* is evaluations of security and performance.

Cost evaluations of known attacks should be quantified. In recognition of the limited evaluation time and the need for clarity regarding the security goals, NIST should pick a *very short* list of *clearly defined* cost metrics for attacks. Comparisons of security-performance tradeoffs should use standard scientific scatterplots. “Categories” have damaged this process and should be scrapped.

For lattice systems in particular, there are huge error bars in current estimates of the cost of known attacks. Consider again the round-3 Kyber presentation saying that the fastest attack known against `kyber512` uses somewhere between $2^{135.5}$ and $2^{165.5}$ “gates”—never mind the question of exactly which “gates” are allowed. These error bars should be included in security comparisons between lattice systems and other systems. Note that “categories” damage this process too: they have far too low precision to communicate error bars.

For comparing security-performance tradeoffs among lattice KEMs, whether one takes a low, medium, or high “gate” count for attack cost, it *seems* that the results are highly correlated with Core-SVP. Lattice-comparison scatterplots that use Core-SVP do not obviously need to be discarded. However, it would be better to upgrade from Core-SVP to a more realistic estimate. NIST should designate a shared cost-estimation method to be used consistently across all lattice submissions.