From: pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent: Friday, May 29, 2020 8:16 PM
To: pqc-comments@list.nist.gov
Cc: pqc-forum
Subject: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER
Attachments: signature.asc

I've been writing a very careful analysis of the cost of a hybrid attack against Kyber-512. As part of this, I've been reviewing the security claims made in the Kyber submission document regarding Kyber-512. I'm filing this comment because

* I'm unable to verify and
* I'm able to disprove parts of

the submission's argument that Kyber-512 meets category 1, the minimum security requirement in NIST's call for submissions, against the attacks that were already described in the submission document.

Rules: Category 1 says that every attack "must require computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit key (e.g. AES128)". This requirement must be met in "_all_ metrics that NIST deems to be potentially relevant to practical security". One of the metrics already mentioned in the call is "classical gates"; NIST estimates that AES-128 key search uses about 2^143 "classical gates".

Consequently, to comply with the minimum security requirements, each parameter set needs each known attack to use "comparable to or greater than" 2^143 "classical gates", no matter what happens in other metrics (such as security against quantum attacks, which I originally thought was the point of post-quantum cryptography---silly me!).

Procedurally, NIST has stated "We're not going to kick out a scheme just because they set their parameters wrong. ... We will respond to attacks that contradict the claimed security strength category, but do not bring the maturity of the scheme into question, by bumping the parameter set down to a lower category, and potentially encouraging the submitter to provide a higher security parameter set." It's obviously important here whether parameters can survive in category 1---this makes the difference between bumping the parameters down and eliminating them.

Attacks against Kyber-512: The starting issue here is already clear in Table 4 in the submission document. The popular "Core-SVP" machinery for claiming security levels, applied to Kyber-512, claims pre-quantum security 2^112 against primal attacks (more precisely, 2^112.24 from beta=385, evidently considering only multiples of 5; my recalculation reaches 2^111.252 from beta=381 with the most popular formulation of Core-SVP, or 2^111.544 from beta=382 with the second most popular) and pre-quantum security 2^111 against dual attacks.

It's also clear how the submission document responds to this problem, namely by disputing the accuracy of Core-SVP. Page 21 of the submission document, under "Gates to break AES vs. core-SVP hardness", gives five brief arguments that BKZ-beta, the main bottleneck in these attacks, is much more expensive than claimed by Core-SVP. The document says "it seems clear that in any actual implementation of an attack algorithm the product of the cost of those items will exceed 2^30".

Multiplying 2^30 by 2^111 gives 2^141. Perhaps this is close enough to count as "comparable" to 2^143. Note that this response does not claim _any_ security margin (despite other provisions in the call about being "conservative in assigning parameters to a given category"), so a mistake at any point can easily be fatal to the conclusion.

The general structure of this response makes sense, but the details don't. Two of the five stated arguments simply don't apply, and I don't see how the other three arguments reach the claimed 2^30.

Inapplicable arguments: The submission document says that Core-SVP ignores "the additional rounding noise (the LWR problem, see [13, 8]), i.e. the deterministic, uniformly distributed noise introduced in ciphertexts via the Compress_q function".

The document says, however, that the standard Kyber-512 attack uses 410 samples. That's a key-recovery attack, making the ciphertext rounding irrelevant. Similarly, the reported Kyber-768/Kyber-1024 attacks are key-recovery attacks. So this argument is incorrect.

The document also says that Core-SVP ignores "the cost of access into exponentially large memory". This doesn't help Kyber-512 close the gap to 2^143 "classical gates": "classical gates" ignore communication costs.

Of course, real attacks can't simply ignore communication costs. For example, counting "classical gates" allows memory-intensive operations such as sorting in an essentially linear number of gates, whereas the 1981 Brent--Kung area-time theorem says that such operations have AT exponent >=1.5. This Kyber-512 attack would be bottlenecked by access to an insane amount of memory. However, if the attack doesn't have enough "classical gates" then, according to the rules, Kyber-512 doesn't meet category 1.

Before the call for submissions, I objected to "craziness such as unit-'time' access to massive storage arrays". I would love to see NIST announcing that it no longer "deems" this debunked metric "potentially relevant to practical security". But, unless and until NIST makes such an announcement, it seems that---unfortunately---we have to consider the consequences of the "classical gates" metric.

Other arguments: Having two of the five arguments disappear doesn't mean that the "exceed 2^30" conclusion is wrong, so let's look at the other three arguments. The document says that Core-SVP ignores

* "the (polynomial) number of calls to the SVP oracle that are required to solve the MLWE problem";

* "the gate count required for one 'operation'"; and

* "additional cost of sieving with asymptotically subexponential complexity".

Regarding the first argument, the traditional estimate is that BKZ-beta costs 8d times as much as SVP-beta, but https://eprint.iacr.org/2019/089 says its experiments are more than 4 times faster than this. Let's guess an overall cost ratio close to 2^11 here. (Maybe a bit more or less, depending on how many BKZ tours are really required.)

Regarding the second argument, https://eprint.iacr.org/2019/1161 estimates slightly more than 2^12 bit operations per near-neighbor "operation". So, okay, we're up to 2^23, which might seem safely on the path to 2^30, but this presumes that the third argument also works.

The third argument is alluding to the procedure that was used to obtain the 2^(0.292 beta) formula for the number of "operations" in SVP-beta:

* There is an asymptotic analysis, under plausible heuristic assumptions, concluding that the 2016 BDGL SVP-beta algorithm uses (3/2+o(1))^(beta/2) "operations". Here 3/2+o(1) means something that _converges_ to 3/2 as beta goes to infinity, but this says nothing about any specific beta, such as beta=381. Note that the same algorithm with the known "dimensions for free" speedups still costs (3/2+o(1))^(beta/2), with a different o(1).

* There is an unjustifiable replacement of (3/2+o(1))^(beta/2) with (3/2)^(beta/2). For example, this converts the "dimensions for free" speedup factor into speedup factor 1, which is very wrong.

* There is a minor change from 3/2, which is around 2^0.5849625, down to 2^0.584 = 2^(2*0.292). For example, for beta=381, this moves (3/2)^(381/2), which is around 2^111.435, down to 2^111.252.

The issue is the middle step in this procedure, suppressing the o(1).
This fake math is sometimes supplemented with speculation that this is a "lower bound", i.e., that the o(1) is nonnegative, i.e., that the actual
(3/2+o(1))^(beta/2) has "additional cost" beyond (3/2)^(beta/2). But why shouldn't this o(1) be negative? Sure, the calculations in

https://groups.google.com/a/list.nist.gov/d/msg/pqc-forum/VHw5xZOJP5Y/UDdGywupCQAJ

and the more detailed calculations in https://eprint.iacr.org/2019/1161 show some increases beyond (3/2)^(beta/2), but why should we think that these aren't outweighed by the known "dimensions for free" speedups?

Another part of the submission document claims that the "hidden sub-exponential factor is known to be much greater than one in practice"
but this claim (1) mixes up the separate factors considered above---of course the experimental timings see memory costs etc.---and (2) doesn't account for "dimensions for free".

Summary: The Kyber documentation gives five brief arguments that its security level is much higher than claimed by Core-SVP, and says "it seems clear that in any actual implementation of an attack algorithm the product of the cost of those items will exceed 2^30". However,

* one of the arguments (ciphertexts) is simply wrong for these Kyber attacks,

* another argument (memory) ignores the announced rules regarding cost metrics,

* a third argument (the o(1)) still lacks justification and could easily end up _reducing_ Kyber's security, and

* the remaining two arguments aren't enough to rescue Kyber-512 without help.

One could try adding a further argument that Core-SVP is inaccurate in its delta formula, but as far as I know all experiments are consistent with the theory that this inaccuracy _overestimates_ security levels--- there's actually a high chance of success with noticeably smaller beta.
One can also object to the Core-SVP usage of "expected" norms of secrets, but fixing this again looks bad for Kyber-512---none of the secrets have fixed weights, so there's an unnecessarily wide range of norms, which means that even smaller choices of beta frequently work.

Does the Kyber team continue to claim that Kyber-512 meets NIST's minimum security requirements? If so, why?

Followup questions: The quotes above show that the proposal of Kyber-512 (and, more broadly, the Kyber submission's assignment of parameters to security categories) relies on disputing the accuracy of Core-SVP---for example, accounting for the number of SVP-beta calls in BKZ-beta, and accounting for the cost of memory. There's a lot to be said for this approach. Core-SVP is indisputably inaccurate, and it's very easy to believe that the cost of memory makes known attacks against Kyber-512 more expensive than AES-128 key search, so Kyber-512 would qualify as category 1 if NIST's rules weren't forcing consideration of an unrealistic metric.

However, given that the Kyber submission document handles Kyber-512 in this way, I don't understand why other parts of the same document say things like this:

> We choose to ignore this polynomial factor, and rather evaluate only the core SVP hardness, that is the cost of one call to an SVP oracle in dimension b, which is clearly a pessimistic estimation (from the defender's point of view). This approach to deriving a conservative cost estimate for attacks against LWE-based cryptosystems was introduced in [7, Sec. 6]. ...
>
> We follow the approach of [7, Sec. 6] to obtain a conservative lower bound on the performance of both sieving and enumeration for the dimensions that are relevant for the cryptanalysis of Kyber. This approach works in the RAM model, i.e., it assumes that access into even exponentially large memory is free.

Does the submission in fact ignore these costs? If we "ignore this polynomial factor" and assume that "access into even exponentially large memory is free" and disregard the inapplicable ciphertext argument then how does the submission rescue Kyber-512? If we _account_ for these factors then how does the submission argue that the resulting higher cost for sieving is a "lower bound" for enumeration, especially quantum enumeration? (Never mind the more recent posting

> https://simons.berkeley.edu/sites/default/files/docs/14988/20200120-lwe-latticebootcamp.pdf

mentioning an "overshoot" idea to reduce the enumeration exponent.)

This incoherence in the Kyber submission is problematic. It's too easy for readers to see all of these descriptions of Kyber as supposedly using "conservative" assignments of security levels, and to miss how much Kyber-512 is on the bleeding edge. Consider, e.g., NIST IR 8240, which complained that a submission was using "a cost model for lattice attacks with higher complexity than many of the other lattice-based candidates". This might sound like Kyber, which had inserted a factor
>2^30 into its cost model, most importantly a huge factor from "the cost
of access into exponentially large memory"---but, no, NIST was talking about another submission and didn't note this aspect of Kyber.

Other submissions: I've been focusing on Kyber here, but there are other submissions that have proposed parameters with Core-SVP claiming security levels below 2^128. I think this is the complete list:

* NTRU: 2^106 for ntruhps2048509.
* Kyber: 2^111 for kyber512.
* NewHope: 2^112 for newhope512.
* SABER: 2^125 for lightsaber.

(The incentives here are a problem. There are many reasons to think that the efficiency of bleeding-edge parameters will attract attention and reward the submission as a whole, while complaints about security will at worst have _those parameters_ removed, according to NIST's rules. The crude pixelation of security levels into five categories also distracts attention from serious efforts to show efficiency-security tradeoffs.)

The presentations of these parameters differ. NTRU is clear in labeling
ntruhps2048509 as category 1 _only if_ costs of memory access are added.
Kyber, NewHope, and SABER have blanket claims of category 1. Kyber and NewHope both repeatedly claim "conservative" evaluations of security but don't actually use those evaluations in assigning security categories.

Kyber ciphertext size jumps by 48% if these parameters are eliminated.
NewHope ciphertext size jumps by 97%. SABER ciphertext size jumps by 48%, although the 2^125 for lightsaber is quite a bit larger than the
2^111 for kyber512 and the 2^112 for newhope512, so maybe this jump isn't needed. NTRU has a denser parameter space and suffers less from having ntruhps2048509 removed.

The next step up as measured by Core-SVP is NTRU Prime, specifically
2^129 for sntrup653 and 2^130 for ntrulpr653. NTRU Prime disputes the Core-SVP accuracy in much more detail than the other submissions. Like NTRU, NTRU Prime defines replacement metrics to take memory access into account for assigning categories, and has a dense parameter space; unlike NTRU, NTRU Prime avoids proposing any bleeding-edge parameters.

The Core-SVP list continues with 131 for round5n10d, 133 for round5n15d,
136 for ntruhrss701, 145 for ntruhps2048677, 146 for round5n1, 147 for lac128, 148 for frodo640, 153 for sntrup761, 154 for babybear, 155 for ntrulpr761, etc. Each step up here makes it harder to imagine that the standard attack uses fewer "classical gates" than AES-128 key search:
there would need to be more and more severe inaccuracies in Core-SVP.

---Dan

From: pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent: Sunday, May 31, 2020 5:14 PM
To: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER
Attachments: signature.asc

I wrote:
> But why shouldn't this o(1) be negative?

I realize that I should give more quantification behind this question.
The question was already raised in the NTRU Prime submission ("Could the cost formula _overestimate_ security for larger sizes, for example by a subexponential factor that is not visible in small-scale experiments?") but it might not be obvious why this is a reasonable question.

Let's define Gates(beta) as the minimum number of "classical gates"
inside known non-quantum BKZ-beta algorithms that are believed to work with high probability (say >=50%). Let's also define Claim(beta) as the "conservative lower bound" (3/2)^(beta/2) = 2^(0.29248125... beta).

Don't existing analyses indicate that the ratio Gates(beta)/Claim(beta) is superpolynomially, almost exponentially, _smaller_ than 1? The short argument for this is that

* "dimensions for free" seem to save a superpolynomial factor while
* all of the compensating overheads seem to be polynomial.

Here are the overheads in more detail:

* BKZ-beta is typically estimated as costing 8d calls to SVP-beta,where d is the dimension. https://eprint.iacr.org/2019/089 saves afactor around 4 by sharing work across SVP-beta calls---hard to seehow this grows with the dimension, but overall the BKZ/SVP costratio looks like roughly d. Maybe 8 tours aren't enough, but https://eprint.iacr.org/2015/1123 looks like it can prove apolynomial bound.

* The 2016 BDGL algorithm handles SVP-beta with a polynomial number of calls to a near-neighbor search. The literature sometimes says alinear number of calls, although the constant is unclear (this relates to how much preprocessing one can afford with a smaller BKZ and how effective the smaller BKZ is). A polynomial bound (with just LLL preprocessing) should be easy to prove, and the high success probability of the algorithm is a plausible conjecture.

* The near-neighbor search uses a database of size proportional to "the reciprocal of the volume of a spherical cap of angle pi/3". The numbers (calculated with a short Sage script) in

  https://groups.google.com/a/list.nist.gov/d/msg/pqc-forum/VHw5xZOJP5Y/UDdGywupCQAJ

seem to grow as roughly sqrt(beta) (4/3)^(beta/2). It should be possible to prove a polynomial bound times (4/3)^(beta/2).

* https://eprint.iacr.org/2019/1161 looks beyond the database size and counts the number of popcount computations in the whole near-neighbor computation. Here the numbers look like roughly 128 beta (3/2)^(beta/2). It should be possible to prove a polynomial bound times (3/2)^(beta/2), and again the high successprobability of the algorithm is a plausible conjecture.

* Each popcount computation, in turn, takes about 5n "classical gates" plus n "classical gates" for an initial xor. The optimized nin https://eprint.iacr.org/2019/1161 is normally below 2 beta, and the conjectured success probability certainly doesn't rely on n being more than polynomial in beta.

Overall this says that SVP-beta is roughly 256 beta^2 (3/2)^(beta/2) popcounts and that BKZ-beta is roughly 3072 d beta^3 (3/2)^(beta/2) "classical gates" if the linear number of calls is around beta. The exact numbers aren't easy to pin down, but I don't see anything that would be more than a polynomial times (3/2)^(beta/2).

Meanwhile https://eprint.iacr.org/2020/487 plausibly conjectures that the latest "dimensions for free" speedups handle SVP-beta using a polynomial number of sieves in dimension

beta - (log(13/9)+o(1))beta/log beta.

This change in dimension gives a speedup factor

sqrt(3/2)^((log(13/9)+o(1))beta/log beta)

which is asymptotically much bigger than any polynomial: polynomial factors disappear into the o(1). Consequently Gates(beta) is a factor

sqrt(3/2)^((log(13/9)+o(1))beta/log beta)

smaller than the "conservative lower bound" Claim(beta). The 13/9 is more recent than the round-2 submissions but there was already a 2^Theta(beta/log beta) speedup in https://eprint.iacr.org/2017/999.

Am I missing something here? Is there a dispute about "dimensions for free" achieving a superpolynomial speedup? Are one or more of the overheads listed above claimed somewhere to be superpolynomial? Why is Claim(beta) portrayed as a "conservative lower bound" for Gates(beta)?

The picture changes dramatically if one switches from "classical gates" to a cost metric with realistic accounting for memory: this creates an _exponential_ overhead that outweighs "dimensions for free". But this isn't relevant to submissions trying to argue that attacks use as many "classical gates" as AES-128 key search.

At this point I could use fake math to leap from "Gates(beta) is asymptotically below Claim(beta)" to "Gates(381) is below Claim(381)", but of course this wouldn't be justified. Perhaps Gates(381) is billions of times larger than Claim(381)---i.e., comparable to 2^143, the number of "classical gates" for AES-128 key search. This would rescue Kyber-512 (under the more explicit assumptions that Core-SVP is correct regarding 381 being the minimum beta that works for a primal attack, and that dual attacks don't do much better). But where's the analysis calculating Gates(381)?

https://eprint.iacr.org/2019/1161 plausibly tallies popcounts inside near-neighbor search. There will be a few thousand "classical gates" per popcount. Maybe the BKZ/SVP cost ratio ends up around 1000, although this starts getting fuzzier: how many tours are really needed, and what does the sharing in https://eprint.iacr.org/2019/089 mean for beta=381? Even less clear is the number of SVP calls to near-neighbor search. And then the big wildcard is "dimensions for free", where as far as I know every analysis has been either asymptotic or experimental, saying nothing about beta=381. https://eprint.iacr.org/2020/487 has more precise asymptotics than previous work but that's still not the same as a concrete analysis.

Given published analyses, a user who simply wants all _known_ attacks to use as many "classical gates" as AES-128 cannot safely select Kyber-512 (never mind risks of advances in attacks). If the Kyber team has its own private analysis pinning down the uncertainties described above and saying that Gates(381) really is big enough, the analysis should be posted for review by the community. Kyber's claim to be relying on "conservative lower bounds" should be withdrawn in any case.

---Dan

--

From: pqc-forum@list.nist.gov on behalf of Peter Schwabe <peter@cryptojedi.org>
Sent: Thursday, June 4, 2020 11:22 AM
To: pqc-forum; pqc-comments@list.nist.gov
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

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

Dear Dan, dear all,

Thank you for this detailed comment. We agree with most of your analysis, in particular that, as far as we understand, you conclude that
- the security of Kyber-512 exceeds the security of AES-128 in all realistic classical cost models;
- the security of Kyber-512 exceeds the security of AES-128 in all realistic quantum cost models; and
- the current understanding of attacks against lattice schemes is not at the point, yet, where it allows to give extremely precise gate counts that would support or disprove our claim of at least 2^143 for Kyber-512.

Also, we agree that the rounding noise in the ciphertext does not add security against the key-recovery attack we consider; we did not correctly update this bit of the specification when removing the rounding in the public key in our round-2 tweaks. We will update the specification accordingly.

More generally, regarding the value of the core-SVP analysis and the gap to actual attacks, our position is that the core-SVP analysis serves as a conservative lower bound based on the parts of attacks we believe to have a reasonable understanding of at the moment. It is clear that there is a significant gap between core-SVP hardness estimates and actual attacks; it is also clear that the size of this gap is different in different cost metrics. We strongly believe that indeed, further research is required to reach a better understanding of this gap and there is a risk that such research eventually shows that certain parameter sets of various NIST candidates, including Kyber-512, fall slightly short of their respective claims of NIST security levels. We also agree that this risk is maximized when applying the gate-count metric to attacks that require exponentially large memory.

Let us take a closer look at the size of the gap for Kyber-512 in the gate-count metric:
- First, let's take a look at the optimal block size. The round-2 Kyber specification uses β=385; in your comment you use β=381, but very recent software by Dachman-Soled, Ducas, Gong, and Rossi shows that the median successful β during a progressive BKZ would in fact be β=388, with only 0.012 probability of success at β=381 [DDRG20]. Note that this paper reports extreme accuracy of these refined estimates (up to 1 bikz) compared to practice in reachable dimensions. There are several reasons why the 2015 estimates are somewhat inaccurate, and in relevant dimensions the "HKZ-tail-shape" seems to be what dominates these inaccuracies; this tail inaccuracy from the 2015 estimates induces a security *underestimate* for large parameters. The [DDRG20] simulations also accounts for the variable length of the secret. A patch for that software to deal with Kyber-512 is available at

https://github.com/lducas/leaky-LWE-Estimator/blob/Kyber_Refined_Estimate/Sec6_applications/kyber-512.sage

- Second, there is the number of calls to the SVP oracle. We agree with your 2^11 estimate.

- Third, we need to understand the cost (in terms of gate-count) of one call to the SVP oracle or, in other words, obtain an estimate for the o(1) term in the asymptotic complexity. We agree that, at least conceptually, the recent dimensions-for-free technique [Duc17,ADH+19] challenges the o(1) > 0 assumption from 2015 [ADPS15, AGVW17].

We see two reasonable ways to estimate the cost of the SVP oracle.

1. One can extrapolate from experiment. If we (ab)use the fit of [ADH+19] (Fig. 3) and scale the results of that paper to β=388, we obtain a cost of one oracle call of 2^(.296*β + 12), i.e., close to 2^127 CPU cycles. This extrapolation is questionable in both directions: the curve is still convex here, so underestimating extrapolation, but the underlying sieve is BGJ1 [BGJ15] not BDGL [BDGL15], so overestimating extrapolation. In the computation described in [ADH+19], most of the CPU cycles are spent on vectorized floating-point multiply-accumulate and on xor-popcount operations. The translation from CPU cycles to gates is certainly very different for those operations and also dedicated hardware would choose different tradeoffs. A conservative estimate might be 2^5 gates per CPU cycle, a more realistic one is 2^10 and a CPU performing multiple fused floating-point multiply-accumulates goes well beyond that, say 2^12?

2. Alternatively, one can write out the complete attack circuit. A large part of the sieving subroutine has been considered in [AGPS19]. Accounting for dimensions for free, using the optimistic condition from [D17], an SVP-call with β=388 requires a sieve at dimension around 352. For this [AGPS19] predicts a gate cost around 2^130, per call to the FindAllPair function. This function should be called polynomially many times for a sieve, and that number remains to be quantified. This model ignores other subroutines and relies on an idealized model of sieving that, judging by the [ADH+19] implementation, is generous to the attacker (e.g. ignoring collisions).

Now, what do we make of these numbers? Based on 2^113 core-SVP hardness for dimension 388, together with 2^11 calls to the oracle and a cost of
2^12 gates per oracle call we reach a gate count of 2^136. Extrapolating from actual state-of-the-art software [ADH+19] gives us 2^127 CPU cycles per oracle call, which translates to somewhere between 2^132 and
2^139 gates per oracle call and thus a total attack cost of between
2^143 and 2^150. Alternatively, calling the dimension 352 circuit from [AGPS19] a total of 2^11 times leads to a gate count of at least 2^141, with a missing factor (≥ 1) for the number of calls to FindAllPairs in a sieve.

We agree that 136 and 141 are smaller than 143, but at the moment we do not consider this to be a sufficient reason to modify the Kyber-512 parameter set. The additional memory requirement of this attack strongly suggests that Kyber-512 is more secure than AES-128 in any realistic cost model. We are very curious to learn about the position that NIST takes on this and more generally on the importance of the gate-count metric for attacks that require access to large

memories - we are talking about 2^88 bits or so here. For a fun sense of scale: a micro-SD card has a 2^3.5 mm^2 footprint (if you stand it on the short end). A planar sheet of terabyte micro-SD cards the size of New York City (all five boroughs, 800 km^2 ~ 2^49.5 mm^2) would hold 2^89 bits.

All the best,

The Kyber team

[BGJ15] Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search Becker, Gama, Joux
https://eprint.iacr.org/2015/522

[BDGL15] New directions in nearest neighbor searching with applications to lattice sieving Becker, Ducas, Gama, Laarhoven
https://eprint.iacr.org/2015/1128

[ADPS15] Post-quantum key exchange - a new hope Alkim, Ducas, Pöppelmann, Schwabe
https://eprint.iacr.org/2015/1092

[D17] Shortest Vector from Lattice Sieving: a Few Dimensions for Free Ducas
https://eprint.iacr.org/2017/999.pdf

[AGVW17] Revisiting the Expected Cost of Solving uSVP and Applications to LWE Albrecht, Göpfert, Virdia, Wunderer
https://eprint.iacr.org/2017/815

[ADH+19] The General Sieve Kernel and New Records in Lattice Reduction Albrecht, Ducas, Herold, Kirshanova, Postlethwaite, Stevens
https://eprint.iacr.org/2019/089.pdf

[AGPS19] Estimating quantum speedups for lattice sieves Albrecht, Gheorghiu, Postelthwaite, Schanck
https://eprint.iacr.org/2019/1161

[DDRG20] LWE with Side Information: Attacks and Concrete Security Estimation Dachman-Soled, Ducas, Gong, Rossi
https://eprint.iacr.org/2020/292

From: 'Perlner, Ray A. (Fed)' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Tuesday, June 9, 2020 11:39 AM
To: Peter Schwabe; pqc-forum; pqc-comments@list.nist.gov
Subject: RE: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

Dear Kyber team (And Dan and whoever else is listening) Thank you for these details regarding the Kyber team's approach to security estimates.

The Kyber team states:

"We are very curious to learn about the position that NIST takes on this and more generally on the importance of the gate-count metric for attacks that require access to large memories"

We would like to preface our response by noting that all 5 security categories are designed to be well beyond the reach of any current technology that could be employed to implement a computational attack. The reason we distinguish among security levels beyond what's currently feasible is as a hedge against improvements in both technology and cryptanalysis. In order to be a good hedge against technology improvements, a model must be realistic, not just for current technology, but for future technology (otherwise we wouldn't consider quantum attacks at all.) This means that we should be more convinced by hard physical limits than the particular limitations of current technology (although if something is difficult for current technology there often is a fundamental physical reason why, even if it's not obvious.) As a hedge against improvements in cryptanalysis, it's not 100% clear that a realistic model of computation, even for future technology, is optimal in providing that hedge. The more complicated a model of computation is, the harder it is to optimize an attack for that model, and the less certain we can be that we are truly measuring the best attacks in that model. As such, the gate-count metric has the virtue of being simpler and easier to analyze than more realistic models.

With that said, let's assume we are aiming for a realistic model of computation.

There are a number of ways that memory-intensive attacks might incur costs beyond what's suggested by gate count:

First of all, there is the sheer cost of hardware. This is what seems to be alluded to by the Kyber team's observation that

"For a fun sense of scale: a micro-SD card has a 2^3.5 mm^2 footprint (if you stand it on the short end). A planar sheet of terabyte micro-SD cards the size of New York City (all five boroughs, 800 km^2 ~ 2^49.5 mm^2) would hold 2^89 bits."

This sounds quite impressive, but note that if we were to try to perform 2^143 bit operations with current hardware, we could for example invest in high-end bitcoin mining equipment. By our calculations, a planar array of high-end mining boxes could cover New York city 30 times over and still take 10000 years to perform 2^143 bit operations (Assuming you can dissipate all the heat involved somehow.) Moreover, this equipment would need to be powered by an array of solar cells (operating at 20% efficiency) covering all of North America. As such, we are unconvinced based on hardware costs alone that 2^89 bits of memory is enough to push an attack above level 1.

A second way memory could incur costs is latency. A number of submitters have pointed out that information cannot travel faster than the speed of light, and we are inclined to agree. However, some have gone further and suggested that memory must be arranged in a 2-dimensional fashion. We are unconvinced, as modern supercomputers tend to use a meaningfully 3-dimensional arrangement of components (although the components themselves tend to be 2-dimensional.) It's also worth noting that sending data long distance can be done at near light speed, (e.g. via fiber optics), but data travels somewhat slower over short distances in most technology we are aware of.

Finally, there is energy cost. Accessing far-away memory tends to cost more energy than nearby memory. One might in fact argue that what gate cost is really trying to measure is energy consumption. Some submitters have advocated modeling this by using a gate model of computation where only local, nearest-neighbor interactions are allowed. This however, seems potentially too pessimistic, because we would be modeling things like long distance fiber optic connections by a densely packed series of gates. It seems clear that sending a bit over a kilometer of fiber optic, while more expensive than sending a bit through a single gate is less expensive than sending a bit through a densely packed series of gates a kilometer long. There is also at least a logarithmic gate cost in the literal sense, since you need logarithmically many branch points to get data to and from the right memory address. For random access queries to extremely small amounts of data, the cost per bit gets multiplied by a logarithmic factor since you need to send the address of the data a good portion of the way, but the algorithms in question are generally accessing consecutive chunks of memory larger than the memory address, so we can probably only assume that random access queries have a log-memory-size cost per bit as opposed to a log^2-memory-size cost per bit, at least based on this particular consideration.

Overall, we think it's fairly easy to justify treating a random access query for b consecutive bits in a memory of size N as equivalent to a circuit with depth N^(1/3), and using log(N)(b+log(N)) gates. (Assuming that the random-access query can't be replaced with a cheaper local, nearest-neighbor circuit.) We are here using depth to model not just the number of wires, but their length, assuming a 3-dimensional arrangement. This model is almost certainly an underestimate of the true costs of memory, but it's somewhat difficult to justify treating memory as much more expensive than this, without making assumptions about future technology that might be wrong.

So what does that mean for NIST's decisions? We recognize that, given known attacks, lattice schemes like Kyber are most likely to have their security underestimated by the (nonlocal) gate count metric as compared to a more realistic memory model. However, without very rigorous analysis, it is a bit difficult to say by how much. In cases where we think the possible attack space is well explored, and the (nonlocal) gate count of all known attacks can be shown to be very close to that of breaking AES or SHA at the appropriate level, and the attacks in question can be shown to need a lot of random access queries to a large memory, we're currently inclined to give submitters the benefit of the doubt that memory costs can cover the difference. To the extent any of those assumptions do not hold (e.g. if the gate count isn't very close to what it should be ignoring memory costs) we're less inclined. We're planning on doing a more thorough internal review of this issue early in the third round. If we think the security of a parameter set falls short of what it should be, but we still like the scheme, we will most likely respond by asking the submitters to alter the parameters to increase the security margin, or to provide a higher security parameter set, but we would prefer not to have to do this. It is worth noting that even though we do not yet have a final answer as to how it relates to NIST's 5 security categories, we feel that the CoreSVP metric does indicate which lattice schemes are being more and less aggressive in setting their parameters. In choosing third round candidates we will adjust our view of the performance of the schemes to compensate.

As always, we warmly encourage more feedback and community discussion.

Thank you,
NIST PQC team

-----Original Message-----
From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> On Behalf Of Peter Schwabe
Sent: Thursday, June 4, 2020 11:22 AM
To: pqc-forum <pqc-forum@list.nist.gov>; pqc-comments@list.nist.gov
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

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

Dear Dan, dear all,

From: Perlner, Ray A. (Fed)
Sent: Wednesday, June 17, 2020 2:19 PM
To: internal-pqc
Subject: FW: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

This looks like it was meant for pqc-forum, but apparently I'm the only one who got it

-----Original Message-----
From: samuel.jaques <sam@samueljaques.com>
Sent: Wednesday, June 17, 2020 10:21 AM
To: Perlner, Ray A. (Fed) <ray.perlner@nist.gov>
Cc: pqc-forum <pqc-forum@list.nist.gov>; pqc-comments@list.nist.gov
Subject: RE: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

Dear Ray and the NIST PQC team,

I have a few questions about this.

First, does heat dissipation force a two-dimensional layout? You mention that today's supercomputers are three-dimensional, but they are also quite flat: It looks like Summit is about 23m x 23 m x 2m. Oak Ridge's website also claims that Summit uses 15,000 L of water every minute to cool it. A completely 3-dimensional layout would still have a two dimensional boundary, and so for N computing units, the heat output/area at the boundary would increase with N^1/3. At scales around 2^143, will that be feasible?

Second, when we consider gate cost, do we consider the number of gates in the circuit or the number of gates that are "active" in some way? For example: In this quantum memory https://arxiv.org/abs/0708.1879, a signal enters at one point and each gate directs it one of two directions. If there are N gates the signal only passes through log N gates, so if the signal loses some energy with each gate and we measure cost by total energy lost, the cost is only log N, since most of the gates are "idle".

Alternatively, if we need to "apply" the gates to direct the signal, then we will need to apply every gate, since we don't know which ones the signal will actually propagate through. Thus the cost is proportional to N.

As abstract circuits these are equivalent, but the costs are very different. The second seems like a good model for surface code quantum computers, and the first seems like a common model for classical computers (though I'm very curious about whether this is actually the right way to think about classical computers). Hedging against future technologies probably means imagining more efficient architectures than a surface code, but maybe error rates, superposition, etc., could force us to keep using the second model.

Do you and/or NIST have an opinion on which gate cost is more accurate for quantum computers in the long term?

Finally, you mentioned that you want to hedge against advances in both technology and cryptanalysis. How should we balance these? For example, is it more likely that an adversary will solve all the problems of fault-tolerance, heat dissipation, and scaling to build enormous quantum memory, or that they will discover an incredible lattice attack that quickly breaks the same scheme on consumer hardware? At least to me, it feels inconsistent to hedge against unrealistic

hardware advances but not unrealistic algorithmic advances. On the other hand, the more assumptions we make about an adversary's limitations, the more likely we are wrong, so fewer assumptions could still be safer, even if they are inconsistent.

Best,
Sam

------- Original Message -------
On Tuesday, 9 June 2020 16:39, 'Perlner, Ray A. (Fed)' via pqc-forum <pqc-forum@list.nist.gov> wrote:

> Dear Kyber team (And Dan and whoever else is listening) > Thank you for these details regarding the Kyber team’s approach to security estimates.
>
> The Kyber team states:
>
> “We are very curious to learn about the position that NIST takes on this and more generally on the importance of the gate-count metric for attacks that require access to large memories”
>
> We would like to preface our response by noting that all 5 security categories are designed to be well beyond the reach of any current technology that could be employed to implement a computational attack. The reason we distinguish among security levels beyond what’s currently feasible is as a hedge against improvements in both technology and cryptanalysis. In order to be a good hedge against technology improvements, a model must be realistic, not just for current technology, but for future technology (otherwise we wouldn’t consider quantum attacks at all.) This means that we should be more convinced by hard physical limits than the particular limitations of current technology (although if something is difficult for current technology there often is a fundamental physical reason why, even if it’s not obvious.) As a hedge against improvements in cryptanalysis, it’s not 100% clear that a realistic model of computation, even for future technology, is optimal in providing that hedge. The more complicated a model of computation is, the harder it is to optimize an attack for that model, and the less certain we can be that we are truly measuring the best attacks in that model. As such, the gate-count metric has the virtue of being simpler and easier to analyze than more realistic models.
>
> With that said, let’s assume we are aiming for a realistic model of computation.
>
> There are a number of ways that memory-intensive attacks might incur costs beyond what’s suggested by gate count:
>
> First of all, there is the sheer cost of hardware. This is what seems to be alluded to by the Kyber team’s observation that > > “For a fun sense of scale: a micro-SD card has a 2^3.5 mm^2 footprint (if you stand it on the short end). A planar sheet of terabyte micro-SD cards the size of New York City (all five boroughs, 800 km^2 ~ 2^49.5 mm^2) would hold 2^89 bits.”
>
> This sounds quite impressive, but note that if we were to try to perform 2^143 bit operations with current hardware, we could for example invest in high-end bitcoin mining equipment. By our calculations, a planar array of high-end mining boxes could cover New York city 30 times over and still take 10000 years to perform 2^143 bit operations (Assuming you can dissipate all the heat involved somehow.) Moreover, this equipment would need to be powered by an array of solar cells (operating at 20% efficiency) covering all of North America. As such, we are unconvinced based on hardware costs alone that 2^89 bits of memory is enough to push an attack above level 1.
>
> A second way memory could incur costs is latency. A number of submitters have pointed out that information cannot travel faster than the speed of light, and we are inclined to agree. However, some have gone further and suggested that memory must be arranged in a 2-dimensional fashion. We are unconvinced, as modern supercomputers

From: 'Perlner, Ray A. (Fed)' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Friday, June 19, 2020 4:54 PM
To: Sam Jaques; pqc-forum
Cc: peter@cryptojedi.org; pqc-comments@list.nist.gov
Subject: RE: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

Hi Sam.

First, it's worth noting that when we defined the security strength categories we did it in such a way that, if two plausible cost models disagree whether an attack costs more than key search on AES-128, the attack is still a valid attack against category 1. This very much puts the burden of proof on those who intend to argue a parameter set is category 1 secure, despite having an attack which, according to simple gate count, costs less than key search against AES-128.

Regarding your specific questions:

"First, does heat dissipation force a two-dimensional layout? You mention that today's supercomputers are three-dimensional, but they are also quite flat: It looks like Summit is about 23m x 23 m x 2m. Oak Ridge's website also claims that Summit uses 15,000 L of water every minute to cool it. A completely 3-dimensional layout would still have a two dimensional boundary, and so for N computing units, the heat output/area at the boundary would increase with N^1/3. At scales around 2^143, will that be feasible?"

This question seems to be assuming the same scaling for memory and active processors. In particular if we assume the energy cost of a random access memory query scales logarithmically with the size of the memory, then there is no obvious heat dissipation problem with a computing system where the memory size scales cubically with the dimension and the number of active processors scales quadratically (less a logarithmic factor if random access memory queries represent a large fraction of the required operations.) Given that the original question had to do with the cost of memory specifically, this seems already enough to cast doubt on the claim that a 2-dimensional nearest neighbor circuit will always model the cost of memory-intensive attacks more accurately than a simple gate model.

I also can't resist adding that, if we assume reversible computation, along the lines of Bennett's Brownian model of computation (as described e.g. here https://www.cc.gatech.edu/computing/nano/documents/Bennett%20-%20The%20Thermodynamics%20Of%20Computation.pdf) it seems quite possible that a 3 dimensional arrangement of processors will also lead to better computational throughput. The idea would be to have cubically many Brownian processors driven by an energy gradient per gate that scales inversely as the square-root of the size of the system. The number of gates per unit time such a system could perform would scale as the 2.5^{th} power of its size, but the heat dissipation would only scale as the 2^{nd} power.

You also ask about quantum memory. Current proposals for quantum computation tend to require active error correction, which seems to imply that just keeping a qubit around for a fixed unit of time incurs a significant computational cost, which would imply something like a memory times depth cost model. I'm aware of some speculative proposals like computing with passively fault tolerant Fibonacci anyons that might be able to get around this particular problem in the long run. Even this, though, seems to be in a model where gates are "applied" rather than sitting around "idle" most of the time. As such, there probably is a better case for treating quantum random access memory as being very expensive, as compared to classical random access memory. In any event, I'm not convinced how we cost quantum memory will end up mattering all that much, given that all the candidates still in the process aren't subject to any known non-generic quantum attacks. There are reasons other than memory costs (e.g. those outlined in https://arxiv.org/abs/1709.10510) for thinking memory intensive generic quantum algorithms like the Brassard-Hoyer-Tapp collision search algorithm won't be better in practice than classical algorithms for the same. Even ignoring that,

parameters targeting categories 1,3, and 5 are almost certainly not going to be subject to a generic quantum speedup better than Grover's algorithm applied to AES if they have enough classical security, and parameters targeting categories 2 and 4 are almost certainly not going to be subject to a quantum attack more memory intensive than the quantum algorithms for collision search. In any event, I wouldn't expect us at NIST to voice a collective opinion on the correct cost model for quantum memory unless we're convinced it matters for some particular parameter set of some particular scheme still in the process.

Finally you asked about the correct model for hedging against cryptanalysis. This is a bit of a squishy question since it's really trying to parametrize our lack of mathematical knowledge, so I can only offer vague guidelines. As I said in the previous email, it seems like simpler is better for maximizing the chances that we'll actually get the cryptanalysis right the first time. I would also add that math doesn't seem to care very much about the energy budget of an earth sized planet or how many atoms are in its crust, so if your claim to have a large security margin relies crucially on exceeding some such physical limit, you might not have as much as you think. We've already had examples of attacks on some schemes that have brought the security of a parameter set from the completely ridiculous category 5 all the way down to category 1 or even lower without contradicting the claim that the scheme is strong in an asymptotic sense.

Best,
Ray

From: Sam Jaques <sam.e.jaques@gmail.com>
Sent: Thursday, June 18, 2020 5:13 AM
To: pqc-forum <pqc-forum@list.nist.gov>
Cc: peter@cryptojedi.org; pqc-comments@list.nist.gov; Perlner, Ray A. (Fed) <ray.perlner@nist.gov>
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

Dear Ray and the NIST PQC team,

I have a few questions about this.

First, does heat dissipation force a two-dimensional layout? You mention that today's supercomputers are three-dimensional, but they are also quite flat: It looks like Summit is about 23m x 23 m x 2m. Oak Ridge's website also claims that Summit uses 15,000 L of water every minute to cool it. A completely 3-dimensional layout would still have a two dimensional boundary, and so for N computing units, the heat output/area at the boundary would increase with N^1/3. At scales around 2^143, will that be feasible?

Second, when we consider gate cost, do we consider the number of gates in the circuit or the number of gates that are "active" in some way? For example: In this quantum memory (https://arxiv.org/abs/0708.1879), a signal enters at one point and each gate directs it one of two directions. If there are N gates the signal only passes through log N gates, so if the signal loses some energy with each gate and we measure cost by total energy lost, the cost is only log N, since most of the gates are "idle".

Alternatively, if we need to "apply" the gates to direct the signal, then we will need to apply every gate, since we don't know which ones the signal will actually propagate through. Thus the cost is proportional to N.

As abstract circuits these are equivalent, but the costs are very different. The second seems like a good model for surface code quantum computers, and the first seems like a common model for classical computers (though I'm very curious about whether this is actually the right way to think about classical computers). Hedging against future technologies probably means imagining more efficient architectures than a surface code, but maybe error rates, superposition, etc., could force us to keep using the second model.

Do you and/or NIST have an opinion on which gate cost is more accurate for quantum computers in the long term?

From: pqc-forum@list.nist.gov on behalf of D. J. Bernstein <djb@cr.yp.to>
Sent: Tuesday, June 23, 2020 4:13 AM
To: pqc-comments@list.nist.gov
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER
Attachments: signature.asc

I've been trying to figure out what the NISTPQC security-evaluation procedures now are, and how they fit with the evaluation procedures specified in the call for proposals. This message emphasizes a yes-or-no question that I think will help clarify the rules.

The call says, for category 1, that every attack "must require computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit key (e.g. AES128)". This requirement applies to "_all_ metrics that NIST deems to be potentially relevant to practical security".

The call summarizes AES-128 key-search costs in a "classical gates"
metric and a depth-limited "quantum gates" metric, while stating that the list of metrics is preliminary. So here's the main question in this
message: Is the same "classical gates" metric still one of the required metrics today, i.e., still one of the metrics "that NIST deems to be potentially relevant to practical security"?

It's important for the community to know what the minimum requirements are for NISTPQC. Anyone evaluating whether an attack speedup kills a parameter set (e.g., whether "dimensions for free" already killed
Kyber-512 before the second round began), or choosing a parameter set in the first place, ends up having to do some sort of cost evaluation; the results often depend heavily on the choice of cost metric. On the flip side, anyone asking whether NISTPQC ensures enough security against real-world attackers also has to know the minimum requirements.

(Subsidiary question relevant to some submissions: It's clear that the minimum requirement is covering attacks against single-target IND-CCA2 when submissions claim IND-CCA2 security, but it's not clear how this accounts for an attack with success probability p. Considering only p=1 is too weak, and considering arbitrarily small p is too strong. Will the rule be "match AES for each p"---basically, multiply cost by 1/p for pre-quantum security, or by 1/sqrt(p) for post-quantum security?)

If the answer to the main question is yes, i.e., that "classical gates"
are still a required metric, then I don't see how Kyber-512 can survive.
The latest gate-count estimates from the Kyber team (9 Jun 2020 15:39:09
+0000) are as low as 2^136. That's not "comparable to or greater than"
2^143; if this is a point of dispute then I think the words "comparable"
and "floor" in the call need quantification for clarity.

There are so many question marks in the analyses of the cost of lattice attacks that 2^136 could easily be an underestimate of the number of gates required (as the same message suggested)---but it could also be an overestimate. Rewarding lattice submissions for the uncertainty would be

* out of whack with the call saying "All submitters are advised to be
  somewhat conservative in assigning parameters to a given category";

* unfair to submissions where the cost of known attacks is better

understood; and

* creating an unnecessary risk of allowing parameter sets where known
attacks are already below the minimum required security level.

The call also asked for category choices to be "especially conservative"
when "the complexity of the best known attack has recently decreased significantly, or is otherwise poorly understood."

On the other hand, the picture can change if the cost metric changes. I recommend scrapping the "classical gates" requirement, for reasons that I already explained in 2016 before this requirement appeared in the call for proposals. At this point I recommend announcing _before_ round 3 that the requirement is gone. Then, for the full round 3, everyone doing security analyses of the parameters for the remaining submissions can focus on what's really required for the NISTPQC standards.

It would have been good to scrap the requirement earlier. I didn't see a huge influence in round 1 of quantifying performance or security; but round 2 has featured more quantitative comparisons, presumably with more influence, and it isn't reasonable to expect that the comparison picture will be robust against changes in security metrics.

(Maybe Core-SVP gets the right ordering of security levels for lattice parameters, but why should we believe this? Core-SVP ignores hybrid attacks, ignores weaker keys, ignores weaker ciphertexts, ignores the security benefit of ciphertext compression for attacks beyond n samples, etc. Furthermore, most of the correlation between Core-SVP and reality seems to come simply from people aiming for a spread of security levels, but what we usually want to know is _security vs. performance_. There are several candidates that are so close in security vs. performance that Core-SVP can't give a straight answer regarding the order---see the graphs in https://cr.yp.to/papers.html#paretoviz. It's easy to see how changing the metric can change the picture. Also, Core-SVP is a mechanism for claiming security levels _for lattice systems_, while we've already seen various quantitative comparisons between lattice systems and non-lattice systems.)

Starting from this background, I've been reading carefully through NIST's latest messages, in the hope of seeing a clear statement that the "classical gates" requirement is being eliminated. I see various comments that sound like portions of rationales for various rules, but only a few comments about what the rules are, and those comments are puzzling. For example, here's one of the comments:

This very much puts the burden of proof on those who intend to argue
a parameter set is category 1 secure, despite having an attack which,
according to simple gate count, costs less than key search against
AES-128.

At first glance this sounds like it's reiterating the "_all_ metrics"
rule---but wait a minute. If "classical gates" are one of the "metrics that NIST deems to be potentially relevant to practical security" then the call says that these parameters do _not_ qualify for category 1, so how is anyone supposed to "argue" that they _do_ qualify? The statement above sounds like "This very much puts the burden of proof on parameter sets that don't meet the requirement to argue that they do meet the requirement"---which doesn't make sense. If the goal was instead to say that it's okay to flunk the "classical gates" requirement if one instead argues security S, then what is this replacement goal S?

It's even more puzzling to see the following comment:

In cases where we think the possible attack space is well explored,
and the (nonlocal) gate count of all known attacks can be shown to be
very close to that of breaking AES or SHA at the appropriate level,
and the attacks in question can be shown to need a lot of random
access queries to a large memory, we’re currently inclined to give

submitters the benefit of the doubt that memory costs can cover the difference.

Structurally, this is saying that submissions meeting conditions T, U, and V are no longer required to follow the "classical gates" rule---so different submissions now have different lists of "metrics potentially relevant to practical security", and different definitions of category 1. Compare this to what the call said:

> 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. 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.
>
> 2) To allow NIST to make prudent future decisions regarding when to transition to longer keys.
>
> 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.
>
> 4) To better understand the security/performance tradeoffs involved in a given design approach.

Regarding goal 2, it's standard for a cryptographic security claim "no attacker can break cryptosystem C" to be factored into two separately reviewed claims:

> (1) "no attack with cost <X breaks cryptosystem C";
> (2) "no attacker can afford cost X".

For example, the claim "no attacker can violate C's IND-CCA2 security with probability >=50%" is factored into

> (1) "no attack with cost <X can violate C's IND-CCA2 security with probability >=50%" (this is a question about the performance of algorithms to attack C) and
>
> (2) "no attacker can afford cost X" (this has nothing to do with C; it's a question of analyzing resources).

Specifying the number X and the underlying cost metric cleanly separates two risk analyses relying on different areas of expertise: reviewer 1 analyzes risks of algorithms running faster than we thought, while reviewer 2 analyzes risks of attackers having more resources than we thought. Even _with_ this separation, both parts of the analysis are error-prone, and I worry about anything that complicates the interface between the parts. For example:

> * Consider the fast attacks in https://sweet32.info and https://weakdh.org, both of which exploited easily avoidable communication failures between people analyzing attack algorithms and people choosing primitives. In both cases one can also blame

NSA for pushing users into deploying bleeding-edge cryptosystems,
but it's the community's fault for failing to defend itself.

* Compare https://blog.cr.yp.to/20151120-batchattacks.html to NIST's
dangerous claim that NISTPQC category 1 is "well beyond the reach
of any current technology that could be employed to implement a
computational attack". In context, this claim seems to be trying to
suggest that we shouldn't be worried about allowing parameter sets
that don't even manage to meet category 1.

To limit the risk of error, it's best for NISTPQC to simply have _one_ metric. It's not as good to have security requirements in multiple metrics; and it's even worse to have security requirements in multiple _submission-specific_ metrics. For goal 1 ("meaningful performance
comparisons") it's even more obviously problematic to have submission-specific metrics.

If the answer to my question is that the rules _currently_ require 2^143 "classical gates" but that NIST is _considering_ a change, then I also have comments on the details of various claims that have appeared in this thread regarding the merits of various metrics. However, after the most recent messages, it's not even clear to me that NIST is still following the metrics->requirements structure specified in the call for proposals, never mind the question of which metrics. I would appreciate clarification of what the current rules are.

---Dan

From: 'Perlner, Ray A. (Fed)' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Tuesday, June 23, 2020 6:08 PM
To: Perlner, Ray A. (Fed); D. J. Bernstein; pqc-comments@list.nist.gov
Cc: pqc-forum
Subject: RE: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

Minor clarification: I think the public list that goes to just us NISTers is pqc-comments@nist.gov

Ray

-----Original Message-----
From: 'Perlner, Ray A. (Fed)' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Tuesday, June 23, 2020 5:26 PM
To: D. J. Bernstein <djb@cr.yp.to>; pqc-comments@list.nist.gov
Cc: pqc-forum <pqc-forum@list.nist.gov>
Subject: RE: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

Hi Dan,

You ask whether "the rules _currently_ require 2^143 'classical gates' but that NIST is _considering_ a change". I think this is fairly close to what NIST's position is and has been from the beginning of the NIST PQC standardization process. That is to say that we currently believe classical gate count to be a metric that is "potentially relevant to practical security," but are open to the possibility that someone might propose an alternate metric that might supersede gate count and thereby render it irrelevant. In order for this to happen, however, whoever is proposing such a metric must at minimum convince NIST that the metric meets the following criteria:

1) The value of the proposed metric can be accurately measured (or at least lower bounded) for all known attacks (accurately mere means at least as accurately as for gate count.)
2) We can be reasonably confident that all known attacks have been optimized with respect to the proposed metric. (at least as confident as we currently are for gate count.)
3) The proposed metric will more accurately reflect the real-world feasibility of implementing attacks with future technology than gate count -- in particular, in cases where gate count underestimates the real-world difficulty of an attack relative to the attacks on AES or SHA3 that define the security strength categories.
4) The proposed metric will not replace these underestimates with overestimates.

Meeting these criteria seems to us like a fairly tall order, but feel free to try. If you or anyone would like to propose an alternate metric to replace gate count, we encourage you send your suggestion to us at "pqc-comments@list.nist.gov" or discuss it on the forum.

Best,
NISTPQC team

-----Original Message-----
From: pqc-forum@list.nist.gov <pqc-forum@list.nist.gov> On Behalf Of D. J. Bernstein
Sent: Tuesday, June 23, 2020 4:13 AM
To: pqc-comments@list.nist.gov
Cc: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Tuesday, June 23, 2020 6:35 PM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER
Attachments: signature.asc

I still don't understand what the current rules are.

I understand the latest message to be saying that "classical gates" is still one of the required metrics---thanks for the clarification. I also understand that four specific conditions need to be met to change this, and that NIST views meeting these conditions as a "fairly tall order".

How, then is it possible for NIST to be saying that it is "inclined" to make exceptions, for some submissions, to the minimum required number of "classical gates"? Here's the puzzling quote again:

> In cases where we think the possible attack space is well explored,
> and the (nonlocal) gate count of all known attacks can be shown to be
> very close to that of breaking AES or SHA at the appropriate level,
> and the attacks in question can be shown to need a lot of random
> access queries to a large memory, we're currently inclined to give
> submitters the benefit of the doubt that memory costs can cover the
> difference.

I would instead expect the announced minimum security requirements to be applied to all submissions. (I'm reminded of a Lincoln quote: "The best way to get a bad law repealed is to enforce it strictly.")

---Dan

From: Perlner, Ray A. (Fed)
Sent: Wednesday, June 24, 2020 8:57 AM
To: D. J. Bernstein; pqc-comments
Cc: pqc-forum
Subject: RE: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

Dan,

In your last email you said

" If the answer to my question is that the rules _currently_ require 2^143 "classical gates" but that NIST is _considering_ a change, then I also have comments on the details of various claims that have appeared in this thread regarding the merits of various metrics."

What are those comments? I think sharing them will be far more productive than asking for continued clarification of NIST's subjective opinion of whether we will be convinced by hypothetical arguments we haven't heard yet.

-Ray

-----Original Message-----
From: D. J. Bernstein <djb@cr.yp.to>
Sent: Tuesday, June 23, 2020 6:35 PM
To: pqc-comments <pqc-comments@nist.gov>
Cc: pqc-forum <pqc-forum@list.nist.gov>
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

I still don't understand what the current rules are.

I understand the latest message to be saying that "classical gates" is still one of the required metrics---thanks for the clarification. I also understand that four specific conditions need to be met to change this, and that NIST views meeting these conditions as a "fairly tall order".

How, then is it possible for NIST to be saying that it is "inclined" to make exceptions, for some submissions, to the minimum required number of "classical gates"? Here's the puzzling quote again:

> In cases where we think the possible attack space is well explored,
> and the (nonlocal) gate count of all known attacks can be shown to be
> very close to that of breaking AES or SHA at the appropriate level,
> and the attacks in question can be shown to need a lot of random
> access queries to a large memory, we’re currently inclined to give
> submitters the benefit of the doubt that memory costs can cover the
> difference.

I would instead expect the announced minimum security requirements to be applied to all submissions. (I'm reminded of a Lincoln quote: "The best way to get a bad law repealed is to enforce it strictly.")

---Dan

From:	D. J. Bernstein <djb@cr.yp.to>
Sent:	Wednesday, June 24, 2020 5:23 PM
To:	pqc-comments
Cc:	pqc-forum
Subject:	Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER
Attachments:	signature.asc

'Perlner, Ray A. (Fed)' via pqc-forum writes:
> What are those comments? I think sharing them will be far more
> productive than asking for continued clarification of NIST's
> subjective opinion of whether we will be convinced by hypothetical
> arguments we haven't heard yet.

I'm asking for clarification of what _today's_ NISTPQC rules are.

My understanding of the call for proposals is that a parameter set broken with fewer than 2^143 "classical gates" is below the minimum NISTPQC security requirements (the "floor" in the call) and has to be replaced by a larger parameter set.

The list of metrics was labeled as preliminary, but I haven't seen NIST announcing a change in the list. On the contrary, NIST has now announced conditions M, N, O, P that all have to be met for a metric to replace the "classical gates" metric, and has described meeting all four conditions as a "fairly tall order". So let's assume the "classical gates" rule is (unfortunately!) locked into stone.

Why, then, did NIST indicate that submissions meeting conditions T, U, V will be given "the benefit of the doubt that memory costs can cover the difference"? This is the really puzzling part. Is there a minimum "classical gates" security requirement today for all parameter sets in all submissions, or not?

This seems directly relevant to Kyber-512 (and perhaps to Kyber more broadly, since Kyber loses 48% in ciphertext size if Kyber-512 is eliminated), as I've explained in detail earlier in this thread.

---Dan

From: Moody, Dustin (Fed)
Sent: Thursday, June 25, 2020 3:30 PM
To: D. J. Bernstein; pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

The NIST PQC Standardization Process is a process for evaluating candidate algorithms to determine which ones are most appropriate for standardization. The Call for Papers specified a set of evaluation criteria. NIST evaluation criteria are guidelines and ways to establish benchmarks, not exhaustive rules.

With respect to security, the CFP also specifically said that:

> "NIST intends to consider a variety of possible metrics, reflecting different predictions about the future development of quantum and classical computing technology. NIST will also consider input from the cryptographic community regarding this question."

It is not the case that a scheme either exactly satisfies our criteria (and is thus still in the running) or it doesn't (and is thus removed from consideration). Other factors can, should, and will be taken into account.

We believe that the process of evaluating the security claims of all candidate algorithms needs to continue and we welcome the community's input on metrics that should be used as part of that evaluation process. We have not and will not specify a set of "rules" that must be used to evaluate the security of every candidate without regard to whether using these "rules" would accurately reflect the level of security of every candidate.

Dustin

From: D. J. Bernstein
Sent: Wednesday, June 24, 2020 5:23 PM
To: pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] ROUND 2 OFFICIAL COMMENT: CRYSTALS-KYBER

'Perlner, Ray A. (Fed)' via pqc-forum writes:
> What are those comments? I think sharing them will be far more
> productive than asking for continued clarification of NIST's
> subjective opinion of whether we will be convinced by hypothetical
> arguments we haven't heard yet.

I'm asking for clarification of what _today's_ NISTPQC rules are.