From: Perlner, Ray (Fed)
Sent: Thursday, March 29, 2018 12:13 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: SIKE

Hi submitters and all,

I have a comment regarding the quantum security estimates for SIKE, which may also be of some general interest regarding how quantum attacks are evaluated. In particular, based on the known attacks, it seems to me that the parameters submitted as security strength 1 and 3 should probably be regarded as security strength 2 and 4 respectively.

The quantum security estimate is based on applying Tani's claw finding algorithm (https://arxiv.org/abs/0708.2584) to find a collision between two different functions on domains of size p^(1/4). The algorithm is given for a quantum random access machine, and it requires a memory of size O(p^(1/6)) and O(p^(1/6)) oracle queries (when the queries are performed in series.) As such, the algorithm seems to have the same memory complexity, query complexity, and parallelizability as the Brassard-Hoyer-Tapp(BHT) algorithm (https://arxiv.org/abs/quant-ph/9705002) applied to finding a collision on a hash function with output size log_2(p^(1/2)). I am personally fairly convinced by analyses such as (https://cr.yp.to/hash/collisioncost-20090517.pdf , https://arxiv.org/abs/1207.2307 , and my own https://arxiv.org/abs/1709.10510) that the quantum algorithm for collision search is no better than the best classical algorithms in any physically plausible model of computation. But whether you buy that or not, it seems like direct comparison between Tani and BHT would be enough to justify the claim that breaking SIKE503, for example, with known attacks is no easier than finding collisions in SHA256.

Please let me know if there are any errors in the above analysis. Is there some model of computation I'm not thinking of where, for example, the known attacks on SIKE503 are appreciably cheaper than the known collision attacks on SHA256?

Thanks,
Ray

P.S. The submission gives a detailed analysis of the number of gates involved in the query portion of the computation, and it looks correct as far as I can tell. However, the estimate does not include gates for memory access. In the gate model proper, querying a memory of size p^(1/6) in series p^(1/6) times requires O(p^(1/3)) gates, which is already larger than the classical security estimate. (This is one additional reason the NIST postquantum call for proposals does not list a lower estimate for quantum vs classical gate requirements at security strengths 2 and 4.) Security estimates that ignore the gate cost of quantum memory access are probably better described as using the quantum random access machine model of computation. There is some theoretical justification for thinking that quantum memory access may be cheaper than is implied by the gate model and more in line with the quantum random access machine model (https://arxiv.org/abs/0708.1879). Of course, unlike the case of classical computers (where access to a terabyte memory clearly does not cost a trillion times as much as an ordinary CPU instruction), it is somewhat controversial whether an efficient quantum RAM can be achieved in practice – the needs of fault tolerance suggest that in any near to medium term quantum computing technology, you're going to need to be continuously performing physical gates on memory qubits while they're at rest anyway.

From: David Jao <djao@math.uwaterloo.ca>
Sent: Friday, March 30, 2018 5:24 PM
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE

Hi Ray,

I believe you are correct. Under current knowledge the parameter sets SIKEp503 and SIKEp751 satisfy respectively security categories 2 and 4 in addition to 1 and 3, even though we did not state so in the submission. The claw-finding problem is directly comparable to the problem of finding a hash collision. Thanks for the clarification.

-David

On 2018-03-29 12:13 PM, Perlner, Ray (Fed) wrote:

> Hi submitters and all,
>
> I have a comment regarding the quantum security estimates for SIKE, which may also be of some general interest regarding how quantum attacks are evaluated. In particular, based on the known attacks, it seems to me that the parameters submitted as security strength 1 and 3 should probably be regarded as security strength 2 and 4 respectively.
>
> The quantum security estimate is based on applying Tani's claw finding algorithm (https://arxiv.org/abs/0708.2584) to find a collision between two different functions on domains of size p^(1/4). The algorithm is given for a quantum random access machine, and it requires a memory of size O(p^(1/6)) and O(p^(1/6)) oracle queries (when the queries are performed in series.) As such, the algorithm seems to have the same memory complexity, query complexity, and parallelizability as the Brassard-Hoyer-Tapp(BHT) algorithm (https://arxiv.org/abs/quant-ph/9705002) applied to finding a collision on a hash function with output size log_2(p^(1/2)). I am personally fairly convinced by analyses such as (https://cr.yp.to/hash/collisioncost-20090517.pdf , https://arxiv.org/abs/1207.2307 , and my own https://arxiv.org/abs/1709.10510) that the quantum algorithm for collision search is no better than the best classical algorithms in any physically plausible model of computation. But whether you buy that or not, it seems like direct comparison between Tani and BHT would be enough to justify the claim that breaking SIKE503, for example, with known attacks is no easier than finding collisions in SHA256.
>
> Please let me know if there are any errors in the above analysis. Is there some model of computation I'm not thinking of where, for example, the known attacks on SIKE503 are appreciably cheaper than the known collision attacks on SHA256?
>
> Thanks,
> Ray
>
> P.S. The submission gives a detailed analysis of the number of gates involved in the query portion of the computation, and it looks correct as far as I can tell. However, the estimate does not include gates for memory access. In the gate model proper, querying a memory of size p^(1/6) in series p^(1/6) times requires O(p^(1/3)) gates, which is already larger than the classical security estimate. (This is one additional reason the NIST postquantum call for proposals does not list a lower estimate for quantum vs classical gate requirements at security strengths 2 and 4.) Security estimates that ignore the gate cost of quantum memory access are probably better described as using the quantum random access machine model of computation. There is some theoretical justification for thinking that quantum memory access may be cheaper than is implied by the gate model and more in line with the quantum

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Sunday, April 01, 2018 12:47 AM
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE
Attachments: signature.asc

Two comments on sub-issues broader than SIKE. (No dispute regarding the general conclusion that known SIKE attacks have basically the same speed as collision attacks.)

Perlner, Ray (Fed) writes:
> There is some theoretical justification for thinking that quantum
> memory access may be cheaper than is implied by the gate model and
> more in line with the quantum random access machine model
> (https://arxiv.org/abs/0708.1879).

There are at least two flaws in the performance analysis in that paper, beyond the issue that you note of having to pay for error correction for stored qubits.

The first flaw looks fatal: https://arxiv.org/abs/1502.03450 analyzes errors in the computations used inside a plausible implementation model for qRAM, and concludes that the cost of correcting these errors makes the qRAM idea lose "its primary advantage".

The second flaw is independently fatal: the 2007 qRAM paper's analysis fails to account for the cost of long-distance communication. Nobody has found a physical communication architecture that beats a two-dimensional nearest-neighbor mesh by more than a constant factor.

> unlike the case of classical computers (where access to a terabyte
> memory clearly does not cost a trillion times as much as an ordinary
> CPU instruction)

The goal of easy programming leads CPU manufacturers to artificially limit their parallelism. This is why CPUs are continually beaten by GPUs at computational tasks, and this also means that to see scalability it's better to look at GPUs than at CPUs. Here are two examples (not top of the line but near top price-performance ratio) showing the GPU trends:

* An NVIDIA GTX 280 from ten years ago had 1.4 billion transistors and 240 ALUs running at 1.296 GHz, with 1GB RAM running at 141 GB/s, i.e., 0.453 bytes per ALU-cycle.

* An NVIDIA GTX 1070 Ti from a few months ago has 7.2 billion transistors and 2432 ALUs running at 1.607 GHz, with 8GB RAM running at 256.2 GB/s, i.e., 0.066 bytes per ALU-cycle.

These 2432 ALUs contain 2432 multipliers, each performing thousands of bit operations every cycle. Random access to memory is much slower, and is becoming more and more of a problem as the chips grow.

Of course these examples don't say that accessing 1TB is a _trillion_ times as expensive as an arithmetic operation. The actual scalability has been understood for 40 years: for example, sorting N small items on a sensibly optimized machine of area N^(1+o(1)) takes time N^(1/2+o(1)), which is N^(1/2+o(1)) times as expensive as doing N small parallel computations. (These exponents are the same for quantum computation and non-quantum computation, although presumably the constant factors will be quite different.)

The reason that these GPU examples show _worse_ than the predicted sqrt scalability in RAM speed is that the GPU's connection to RAM is essentially one-dimensional. This was criticized on slides 43--46 of https://cr.yp.to/talks/2013.06.19/slides-djb-20130619-a4.pdf, before Intel's first announcement of a two-dimensional mesh on its largest Knights Landing chips. Again the 40-year-old models show their predictive value.

---Dan

--

From: francisco@cs.cinvestav.mx
Sent: Tuesday, April 03, 2018 9:09 AM
To: Perlner, Ray (Fed)
Cc: pqc-comments; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE

Hi Ray and all,

We have prepared a manuscript that provides a concrete analysis of the hardness of the Computational Supersingular Isogeny (CSSI) problem. The hardness of this problem is the basis for the security of SIDH/SIKE.
The manuscript is available here:
http://cacr.uwaterloo.ca/techreports/2018/cacr2018-03.pdf
and has also been submitted to the IACR eprint.

The classical Meet-In-The-Middle (MITM) attack on CSSI has an expected running time of O(p^{1/4}), but also has O(p^{1/4}) storage requirements.
In our paper, we demonstrate that the van Oorschot-Wiener collision finding algorithm has a lower cost (but higher running time) for solving CSSI, and thus should be used instead of the meet-in-the-middle attack to assess the security of SIDH against classical attacks.

Together with the large memory requirements of Tani's algorithm and the difficulty of parallelizing a Grover quantum search type of attack, we argue that the van Oorschot-Wiener parallel collision search is actually the most powerful attack known on SIDH/SIKE. This observation, along with a careful cost analysis, leads to the conclusion that the 434-bit prime,
p434 = 2^216 * 3^137 - 1 is sufficient for the SIDH/SIKE protocol to achieve a 128-bit security level, against both classical and quantum attacks. This prime is significantly smaller than the prime p751 = 2^372 * 3^239 - 1 that is used in the Costello-Naehrig-Longa (CLN) library for the 128-bit security level. Using the CLN library, we see that p434 yields a speedup by a factor of approximately 4.8 for SIDH/SIKE operations compared to p751.

Best regards,
Francisco [on behalf of the authors]

>
>

> ---------- Forwarded message ----------
> Date: Thu, 29 Mar 2018 16:13:18 +0000
> From:
> To: pqc-comments <pqc-comments@nist.gov>
> Cc: "pqc-forum@list.nist.gov" <pqc-forum@list.nist.gov>
> Subject: [pqc-forum] OFFICIAL COMMENT: SIKE
>
>
> Hi submitters and all,
>
>
>

From: Perlner, Ray (Fed)
Sent: Tuesday, April 03, 2018 3:16 PM
To: francisco@cs.cinvestav.mx
Cc: pqc-comments; pqc-forum@list.nist.gov
Subject: RE: [pqc-forum] OFFICIAL COMMENT: SIKE

Hi Francisco,

I have a couple of questions regarding your claim.

1) The SIKE submission cites the best classical attack, given by https://arxiv.org/pdf/1310.7789.pdf as having a query complexity of O(p^{1/4}) and a space requirement of O(1) (which is better than your claim of O(p^{1/4})). Your paper does not appear to take this improved attack into account. Is this correct?

2) Your evaluation of Tani's claw finding algorithm seems to rely on a paper of Jaques and Schanck, which I cannot find. Can you tell us where to find the paper or otherwise give further details about which "reasonable cost measures" are used to reach the conclusion that "Tani's algorithm is significantly costlier than Grover's search?" Also, do you know if the analysis takes into account parallelized versions of Tani's algorithm that make parallel queries to memory using techniques like those discussed in https://arxiv.org/pdf/1207.2307.pdf ?

Thanks,
Ray

-----Original Message-----
From: francisco@cs.cinvestav.mx [mailto:francisco@cs.cinvestav.mx]
Sent: Tuesday, April 03, 2018 9:09 AM
To: Perlner, Ray (Fed) <ray.perlner@nist.gov>
Cc: pqc-comments <pqc-comments@nist.gov>; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE

Hi Ray and all,

We have prepared a manuscript that provides a concrete analysis of the hardness of the Computational Supersingular Isogeny (CSSI) problem. The hardness of this problem is the basis for the security of SIDH/SIKE.
The manuscript is available here:
http://cacr.uwaterloo.ca/techreports/2018/cacr2018-03.pdf
and has also been submitted to the IACR eprint.

The classical Meet-In-The-Middle (MITM) attack on CSSI has an expected running time of O(p^{1/4}), but also has O(p^{1/4}) storage requirements.
In our paper, we demonstrate that the van Oorschot-Wiener collision finding algorithm has a lower cost (but higher running time) for solving CSSI, and thus should be used instead of the meet-in-the-middle attack to assess the security of SIDH against classical attacks.

From: luca.defeo@gmail.com on behalf of Luca De Feo <luca.defeo@polytechnique.edu>
Sent: Tuesday, April 03, 2018 5:10 PM
To: pqc-comments; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE

> 1) The SIKE submission cites the best classical attack, given by
https://arxiv.org/pdf/1310.7789.pdf as having a query complexity of O(p^{1/4}) and a space requirement of O(1) (which is better than your claim of
O(p^{1/4})).

I don't think we say the best classical attack is the Delfs-Galbraith algorithm.

We say that the Delfs-Galbraith algorithm is the best **generic** algorithm for the supersingular isogeny walk problem (not the SIDH problem, mind you), however such generic algorithms are no better than exhaustive search on the SIKE key space.

We also say:

> Galbraith's meet-in-the-middle approach can be easily adapted to
> attack SIKE in only $O(\sqrt[4]{p})$ operations

however that should be understood as "there is a meet-in-the-middle attack, that vaguely resembles Galbraith's attack".

I am responsible for most of Section 4.1, my apologies if it is not very clear.

We do not say anything explicitly on space requirements in our submission, however the best attack we describe is the same meet-in-the-middle attack as in Francisco's paper.

By the way, thanks Francisco for sharing your work here!

Best,
Luca

From: francisco@cs.cinvestav.mx
Sent: Tuesday, April 03, 2018 5:51 PM
To: Perlner, Ray (Fed)
Cc: pqc-comments; pqc-forum@list.nist.gov
Subject: RE: [pqc-forum] OFFICIAL COMMENT: SIKE

Hi Ray,

Please see my reply below.

Best regards,
Francisco

> Hi Francisco,
>
> I have a couple of questions regarding your claim.
>
> 1) The SIKE submission cites the best classical attack, given by
> https://arxiv.org/pdf/1310.7789.pdf as having a query complexity of
> O(p^{1/4}) and a space requirement of O(1) (which is better than your
> claim of O(p^{1/4})). Your paper does not appear to take this improved
> attack into account. Is this correct?

Besides the comment given by Luca de Feo I would like to add that the Delfs-Galbraith algorithm deals with supersingular elliptic curves defined over GF(p), (not GF(p^2)). Also, it is not concerned with finding an isogeny of a specific degree (e.g. 2^e) as is the case with SIDH/SIKE.

> 2) Your evaluation of Tani's claw finding algorithm seems to rely on a
> paper of Jaques and Schanck, which I cannot find. Can you tell us
> where to find the paper or otherwise give further details about which
> "reasonable cost measures" are used to reach the conclusion that
> "Tani's algorithm is significantly costlier than Grover's search?"
> Also, do you know if the analysis takes into account parallelized
> versions of Tani's algorithm that make parallel queries to memory
> using techniques like those discussed in
>https://arxiv.org/pdf/1207.2307.pdf ?

Our comment is based on the observation that a direct implementation of Tani's algorithm has an associated O(p^(1/6)) space complexity. For a more detailed answer to your question I would refer you to the paper by Jaques and Schanck that should be available pretty soon.

Thanks,
Francisco

From: luca.defeo@gmail.com on behalf of Luca De Feo <luca.defeo@polytechnique.edu>
Sent: Tuesday, April 03, 2018 6:45 PM
To: pqc-comments; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE

Hi,

> Besides the comment given by Luca de Feo I would like to add that the
> Delfs-Galbraith algorithm deals with supersingular elliptic curves
> defined over GF(p), (not GF(p^2)).

Not completely accurate. Delfs-Galbraith contains two algorithms:

1) An O(p^¼)-time / O(1)-space algorithm for finding isogeny walks between supersingular curves over GF(p).

2) An O(p^½)-time / O(1)-space algorithm for finding isogeny walks between supersingular curves over GF(p^2). This algorithm uses the previous one as a subroutine.

Best,
Luca

From: Sam Jaques <sam.e.jaques@gmail.com>
Sent: Wednesday, April 04, 2018 3:10 PM
To: Perlner, Ray (Fed)
Cc: francisco@cs.cinvestav.mx; pqc-comments; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE

Hello Ray,

An algorithm that can solve a generic claw finding problem for a space of size of 2^n can be used to find collisions for any random hash function on a space of size 2^{2n}. Hence, as you say, breaking SIKE503 via generic claw finding is as hard as finding SHA256 collisions, regardless of similarities to BHT.

In the paper you reference with parallel queries, the computational model assumes every qubit of memory can also act as a processor. So for SIKE503, the 2^83 memory implies 2^83 processors. If each one was running Grover's algorithm instead of Tani's algorithm, in the 2^83 time that Tani's algorithm would take, these processors could search any space of size 2^251.

In fact the computational model in that paper is extremely powerful: Section 5.4 gives an algorithm that would find collisions for an n-bit hash function in time and memory 2^{n/4}.

Our paper, in progress, examines models of quantum computation with increasing computational power/decreasing realism; we have to get very close to a simple query model before Tani's claw finding costs less than Grover's algorithm.

Sam Jaques

On Tue, Apr 3, 2018 at 3:16 PM, Perlner, Ray (Fed) <ray.perlner@nist.gov> wrote:

> Hi Francisco,
>
> I have a couple of questions regarding your claim.
>
> 1) The SIKE submission cites the best classical attack, given by https://arxiv.org/pdf/1310.7789.pdf as having a query complexity of O(p^{1/4}) and a space requirement of O(1) (which is better than your claim of O(p^{1/4})). Your paper does not appear to take this improved attack into account. Is this correct?
>
> 2) Your evaluation of Tani's claw finding algorithm seems to rely on a paper of Jaques and Schanck, which I cannot find. Can you tell us where to find the paper or otherwise give further details about which "reasonable cost measures" are used to reach the conclusion that "Tani's algorithm is significantly costlier than Grover's search?" Also, do you know if the analysis takes into account parallelized versions of Tani's algorithm that make parallel queries to memory using techniques like those discussed in https://arxiv.org/pdf/1207.2307.pdf ?
>
> Thanks,
> Ray
>
> -----Original Message-----
> From: francisco@cs.cinvestav.mx [mailto:francisco@cs.cinvestav.mx]
> Sent: Tuesday, April 03, 2018 9:09 AM
> To: Perlner, Ray (Fed) <ray.perlner@nist.gov>
> Cc: pqc-comments <pqc-comments@nist.gov>; pqc-forum@list.nist.gov

From: Perlner, Ray (Fed)
Sent: Wednesday, April 04, 2018 5:31 PM
To: Sam Jaques
Cc: francisco@cs.cinvestav.mx; pqc-comments; pqc-forum@list.nist.gov
Subject: RE: [pqc-forum] OFFICIAL COMMENT: SIKE

Hi Sam,

Thanks for the clarification.

My concern with your SIKE503 example is that you seem to be comparing a parallel implementation of Grover with a serial implementation of Tani. If we assume, as the SIKE submission does, that Tani's algorithm parallelizes like Grover, then by parallelizing Tani 2^83 ways, we would expect to reduce the time per processor to 2^42. Grover could only search a space of size 2^167 in the same time.

There does seem to be some treatment of parallelizing quantum walk algorithms in the literature (e.g. https://arxiv.org/pdf/1309.6116.pdf) although it will likely need to be adapted slightly to impose a memory requirement other than parallelism times time. Treating claw finding between functions with domain size N as equivalent to element distinctness on a set of size N or 2-sum on a list of size N, the results of the above paper appear consistent with my previous assumption that the time required for a parallelized instance of Tani's algorithm is sqrt (N^2/ M p), where M is memory and p is parallelism (I'm again assuming that M is set equal to parallelism times time.)

Best,

Ray

From: Sam Jaques [mailto:sam.e.jaques@gmail.com]
Sent: Wednesday, April 04, 2018 3:10 PM
To: Perlner, Ray (Fed) <ray.perlner@nist.gov>
Cc: francisco@cs.cinvestav.mx; pqc-comments <pqc-comments@nist.gov>; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE

Hello Ray,

An algorithm that can solve a generic claw finding problem for a space of size of 2^n can be used to find collisions for any random hash function on a space of size 2^{2n}. Hence, as you say, breaking SIKE503 via generic claw finding is as hard as finding SHA256 collisions, regardless of similarities to BHT.

In the paper you reference with parallel queries, the computational model assumes every qubit of memory can also act as a processor. So for SIKE503, the 2^83 memory implies 2^83 processors. If each one was running Grover's algorithm instead of Tani's algorithm, in the 2^83 time that Tani's algorithm would take, these processors could search any space of size 2^251.

In fact the computational model in that paper is extremely powerful: Section 5.4 gives an algorithm that would find collisions for an n-bit hash function in time and memory 2^{n/4}.

Our paper, in progress, examines models of quantum computation with increasing computational power/decreasing realism; we have to get very close to a simple query model before Tani's claw finding costs less than Grover's algorithm.

From: David Jao <djao@math.uwaterloo.ca>
Sent: Thursday, April 05, 2018 9:42 AM
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE

Hi Ray and others,

Regarding your question 1), one point worth highlighting is that Remark
3 in Francisco's paper explains at length the difference between a generic collision search (involving uniformly random functions) and a "golden" collision search (in which the set of collisions has cardinality 1). Large-memory attacks such as meet in the middle work equally well in both settings, but low-memory attacks such as Van Oorschot-Wiener are slower in the latter setting. In our submission, we did not distinguish between these two settings, and assumed (perhaps overly conservatively) that the security level of the former setting applies to the latter setting.

-David

On 2018-04-03 03:16 PM, Perlner, Ray (Fed) wrote:
> Hi Francisco,
>
> I have a couple of questions regarding your claim.
>
> 1) The SIKE submission cites the best classical attack, given by
https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Farxiv.org%2Fpdf%2F1310.7789.pdf&data=02%7C01%7Csara.kerman%40nist.gov%7C0bf3fb9c167f453a316408d59afb177c%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C1%7C636585325583956755&sdata=82cOvKH0zJAqUuwtRdsBe0lqd2Pv000nphicaNwLazU%3D&reserved=0 as having a query complexity of O(p^{1/4}) and a space requirement of O(1) (which is better than your claim of O(p^{1/4})). Your paper does not appear to take this improved attack into account. Is this correct?
>
> 2) Your evaluation of Tani's claw finding algorithm seems to rely on a paper of Jaques and Schanck, which I cannot find. Can you tell us where to find the paper or otherwise give further details about which "reasonable cost measures" are used to reach the conclusion that "Tani's algorithm is significantly costlier than Grover's search?" Also, do you know if the analysis takes into account parallelized versions of Tani's algorithm that make parallel queries to memory using techniques like those discussed in
https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Farxiv.org%2Fpdf%2F1207.2307.pdf&data=02%7C01%7Csara.kerman%40nist.gov%7C0bf3fb9c167f453a316408d59afb177c%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C1%7C636585325583956755&sdata=dP9lRgwIzq3fhUOq3oDNP5B43nbJIxqqdumVk2%2Bw5W4%3D&reserved=0 ?
>
> Thanks,
> Ray
>
>
>
> -----Original Message-----
> From: francisco@cs.cinvestav.mx [mailto:francisco@cs.cinvestav.mx]
> Sent: Tuesday, April 03, 2018 9:09 AM
> To: Perlner, Ray (Fed) <ray.perlner@nist.gov>
> Cc: pqc-comments <pqc-comments@nist.gov>; pqc-forum@list.nist.gov
> Subject: Re: [pqc-forum] OFFICIAL COMMENT: SIKE
>
> Hi Ray and all,
>

From: David Jao <djao@uwaterloo.ca>
Sent: Friday, November 30, 2018 1:57 AM
To: pqc-forum@list.nist.gov
Subject: [pqc-forum] Updated IP statements for SIKE

Dear all,

In the interests of full transparency, we (the SIKE team) wish to announce that we have recently added one new page to the collection of NIST IP statements for SIKE, specifically page 29 (as of this writing) of https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/ip-statements/SIKE-Statements.pdf
(*)

The new page consists of a signed 2.D.2 statement for US patent application 2018/0323973A1, which was recently published by the USPTO.

The following background information may be useful. Unlike most (all?) other submissions, our team has decided to go out of our way to provide, wherever possible, signed 2.D.2 patent releases for patents and patent applications that we control, even when we also believe that those patents and patent applications do not contain any claims that may cover SIKE. This situation applies to:

* US 7499544 (signed 2.D.2 statement on page 28 of (*), and relevant signed 2.D.1 statements on pages 5 and 16 of (*) asserting non-coverage)

* US 2018/0323973A1 (signed 2.D.2 statement on page 29 of (*), and relevant signed 2.D.1 statement on page 26 of (*) asserting non-coverage)

In each case, the 2.D.2 statement is (we believe) redundant, since the patents in question do not cover SIKE. However, in case our belief of non-coverage turns out to be wrong from a legal perspective, we feel that it is better to have the 2.D.2 statement than not to have it.

If anyone wants to discuss specific technical and legal questions of coverage for these particular patents, I would be happy to do so, although I would suggest that for efficiency and noise reduction purposes, specific discussions of individual patents may be best suited for an offline forum with summary reporting to some central clearinghouse such as https://cr.yp.to/patents.html

-David

--