The Case for SIKE

A Decade of the Supersingular Isogeny Problem

Craig Costello*

Abstract. To mark the 10-year anniversary of supersingular isogeny Diffie-Hellman, I will touch on 10 points in defense and support of the SIKE protocol, including the rise of classical hardness, the fact that quantum computers do not seem to offer much help in solving the underlying problem, and the importance of concrete cryptanalytic clarity.
In the final section I present the two SIKE challenges: over 55k USD is up for grabs for the solutions of mini instances that, according to the SIKE team's security analysis, provide significantly less than 64 bits of classical security. I conclude by urging the proponents of other schemes to construct analogous challenge instances.

"SIKE is a fantastic scheme, but its computation is by far the most expensive, and the problem is relatively new. Who knows here?"

– Daniel Apon (NIST) [3].

Introduction

This year marks a decade since Jao and De Feo introduced the supersingular isogeny (SSI) problem [23, Problem 5.1] as a basis for their SIDH post-quantum key exchange protocol. In the 10 years since then, the SSI problem has stood firm in the face of cryptanalytic scrutiny and, as such, SIDH's actively secure incarnation, SIKE [22], remains one of the nine key encapsulation mechanisms that have advanced to Round 3 in NIST's standardisation effort. In fact, the complexity analysis originally used by Jao and De Feo (and in the Round 1 submission of SIKE) has since been shown to be too conservative. Shortly after the Round 1 submission deadline passed, Adj, Cervantes-Vázquez, Chi-Domínguez, Menezes and Rodríguez-Henríquez [1] showed that, in practice, the SSI problem offers more security than originally claimed. Subsequently, the SIKE team reduced their parameter sizes. Recent work by Longa, Wang and Szefer [32] shows that, under a budget-based cost model, these parameters could be reduced even further and still safely exceed the same real-world attack costs of breaking the corresponding AES and SHA-3 instances.

In this white paper I will briefly put forward 10 points in defense and support of the SIKE protocol, addressing some common critiques concerning its suitability for standardisation, as well as highlighting some advantages over its counterparts that are either not mentioned in the SIKE specification, or that warrant further elaboration.

Single sentence summaries. For readers who would prefer a quick skim, there are single sentence summaries at the end of each section that give the gist.

1 A decade unscathed

"Which post-quantum submissions (1) haven't suffered security losses since the #NIST-PQC competition began and (2) are among the 26 submissions in round 2 (which is ending soon)? I think there are exactly 3: SIKE (which scares me for being too new), Classic McEliece, and SPHINCS+."

– Daniel J. Bernstein[1]

*Opinions are my own and are not the views of my employer nor of the SIKE team. Opinions indicate a strong partisan bias, but for what it's worth this comes from being (both in spirit and chronologically) an SIDH fanboy first, and a member of the SIKE team second.

[1] `https://twitter.com/hashbreaker/status/1276449748329750528`.

Let's take a quick look at how some of the famous hard problems in public key cryptography faired in their first 10 years on the scene:

- *DLP.* When they proposed public key cryptography in 1976, Diffie and Hellman cited that discrete logarithms modulo q required $q^{1/2}$ operations. Three years later, Adleman published a subexponential index calculus algorithm [2] that, even on the computers available at the time, would render their initial parameters ($q \approx 2^{200}$) completely broken.
- *Factorisation.* When Rivest, Shamir and Adleman proposed RSA in 1978, they cited Schroeppel's factoring method, arguing that $n \approx 2^{266}$ provides moderate security against current technology, but that using $n \approx 2^{664}$ provides a margin of safety against future developments [43]. In the decade that followed (see [41] for the history and references), Pomerance proposed the quadratic sieve, Lenstra proposed ECM, and finally, in 1988, Pollard circulated a letter describing his idea of factoring numbers which gave birth to the number field sieve (the algorithm that would eventually factor RSA-768 [27]).
- *McEliece.* A bunch of attack papers appeared in the decade following McEliece's 1978 cryptosystem [34]. Some of the most notable papers all appeared in 1988: Lee-Brickell [30], Leon [31], van Tilburg [50] and Stern [45]. As an example, the Lee-Brickell paper gave an attack *"which reduces the work factor significantly (factor of* 2^{11} *for the commonly used example of* $n = 1024$ *Goppa code case)"*. For a system that was originally paramterised to achieve 2^{64} security, this security degradation was significant indeed.
- *NTRU.* A year after it was presented at the CRYTPO rump session in 1996, Coppersmith and Shamir [11] gave improved lattice attacks on NTRU that forced parameters to be increased before the paper was published in 1998 [21] (and attack variants and improvements did not stop there).
- *ECDLP.* Even in the case of the ECDLP, where well-chosen instances have (classically speaking) essentially remained unbroken for almost four decades, Miller's original paper [37, p. 425] specified the example curve $E/\mathbb{F}_p\colon y^2 = x^3 - ax$ with $p \equiv 3 \bmod 4$. He went on to note that it "*may be prudent to avoid curves with complex multiplication (CM) because the extra structure might somehow be used to give a better [ECDLP] algorithm*", however it turned out that it was not the CM that was the problem[2], but rather the supersingularity of E. In the decade that followed the MOV/Frey-Rück attack [35,17] dealt a devastating blow to supersingular curves (in the ECDLP context!).

The point is that it is extremely rare to find a paper proposing a new hard problem for use in public key cryptography where the claimed attack complexity and the proposed concrete parameterisations survive a decade without any security degradation. But this is precisely the state of affairs with Jao and De Feo's SSI problem [23, Problem 5.1] as it stands now in 2021. In fact, as is detailed in the next section, their initial security analysis has proven to be rather conservative and the parameters given in [23, Table 2] offer more concrete security than the asymptotic analysis suggests.

A common critique of SIKE (like Bernstein's above) is that the SSI problem is too new. Of course, when it comes to confidence in a hard cryptographic problem, there is no substitute for the test of time. But it is important to stress that the metric used here should not be how long a problem has been around, but rather for how long no progress has been made in improving the complexity of the best known attack(s). Anyone who is "scared" of a problem that has stood firm for a decade should then be terrified of a problem that has been around for half that long (e.g., the new structured lattice problem underlying NTRU Prime), especially if, by their own account, it has already suffered a security loss.

Summary. The SSI problem may only be 10 years old, but it's off to a better start than any other public key problem I can think of.

[2] Well-chosen non-supersingular curves with CM, e.g. Bitcoin's curve, are fine for use.

2 The rise and rise of classical hardness

> *"So, on average, the 2^{80} storage device will be accessed $2^{48.4}$ times during each unit of time. The cost of these accesses will certainly dominate the computational costs. Thus our security estimates, which ignore communication costs, should be regarded as being conservative."*
>
> – Gora Adj, Daniel Cervantes-Vázquez, Jesús-Javier Chi-Domínguez, Alfred Menezes and Francisco Rodríguez-Henríquez [1]

Shortly after the Round 1 NIST PQC submission deadline, the authors quoted above argued that the security analysis in the SIKE submission was too conservative. They proposed that, rather than using the $O(p^{1/4})$ meet-in-the-middle time *and* space complexities to analyse the classical hardness of the SSI problem, it is the runtime of the van Oorschot-Wiener (vOW) golden collision finding algorithm [49] that is relevant. The smallest of the original SIKE parameters had a 503-bit prime p, and using $p^{1/4} \approx 2^{125}$ *units* of memory would require more than the collective storage resources of the planet. Moreover, NIST defined security levels in accordance with the computational resources required to break symmetric primitives: e.g. for Level 1 security, "*any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit key*" [47, §4.A.5]. A key search on AES-128 requires (according to NIST) 2^{143} classical gates, but such attacks only require a relatively small amount of storage. Thus, the authors of [1] argue that SIKE's security should instead be assessed by fixing an upper bound on the available memory and analysing the runtime of the best known algorithm subject to this bound.

In this case the best-known algorithm becomes vOW: with m processors working in parallel, and with access to w units of storage, the vOW algorithm expects to find a secret isogeny (in a balanced SIKE parameterisation using the prime p) in time

$$\frac{2.5}{m} \cdot \left(\frac{p^{3/8}}{\sqrt{w}} \right) \cdot t, \tag{1}$$

where t is the time required to compute an isogeny of degree $\approx p^{1/4}$. Three separate analyses of the vOW algorithm [1,24,15], all of which took $w \geq 2^{80}$, found that the classical security of SIKEp503 comfortably meets NIST's Level 2 requirements, and that the classical security of SIKEp434 (from [1]) comfortably meets NIST's Level 1 requirements. SIKEp751, which initially targeted NIST's Level 3, was subsequently replaced by SIKEp610 (again from [1]) and was itself bumped up to target Level 5 security.

It should be noted that (i) according to all the data points I could find, 2^{80} units of storage would require more than the current world data storage capacity, and (ii) as the quote from [1] above says, the complexity in (1) assumes communication is free.

A recent paper by Longa, Wang and Szefer [32] applied a budget-based model to analyse the cost of breaking various instances of SIKE, AES and SHA-3. In the case of SIKE, they designed tailored hardware accelerators in order to model an ASIC-powered vOW attack and surveyed the ASIC implementations of AES and SHA-3 from the literature in order to get a real-world cost-based comparison. They concluded that *"This analysis, together with the state-of-the-art quantum security analysis of SIKE, indicates that the current SIKE [Round 3] parameters offer a wide security margin"*. They proposed that the 434-, 610-, and 751-bit primes currently in the SIKE submission could be replaced by the smaller 377-, 546-, and 697-bit primes, respectively, and that attacking the proposed instances would still cost more (or, for a fixed budget, take longer) than the respective attacks on AES128, AES192, and AES256 [32, §6].

Summary. The SIKE parameters have recently decreased because the SSI problem is harder than was initially thought, and this could well happen again.

3 Quantum computers don't really help

> *"An adversary with enough quantum memory to run Tani's algorithm with the query-optimal parameters could break SIKE faster by using the classical control hardware to run van Oorschot–Wiener."*
>
> – Samuel Jaques and John M. Schanck [24]

Just like the classical security story in the previous section, it turns out that Jao and De Feo were also rather conservative in their original quantum security analysis. They cited Tani's claw-finding algorithm [46] as the best quantum algorithm for the SSI problem, which runs in $O(p^{1/6})$ time and is optimal for black-box claw finding attacks [23, §5]. This $O(p^{1/6})$ complexity was in turn used by the SIKE team in their initial submission.

In early 2019, Jaques and Schanck [24] gave a comprehensive analysis into the complexity of known quantum attacks against SIKE, proposing improved models for the analysis of quantum computation in the context of cryptanalysis. Their work showed that the best quantum attacks against SIKE actually require $\tilde{O}(p^{1/4})$ (RAM) operations, and subsequently they increased the security estimates for the SIKE parameters. A more recent paper by Jaques and Schrottenloher [25] gives high memory but low gate algorithms for solving SSI that prompts updated estimates on the quantum attack complexities; in most instances their algorithms *"lower the quantum security of SIKE compared to the results of [24], but not enough to reduce the claimed security levels"*.

On the left "Best quantum" side of Table 1 is the lowest quantum complexity from the four complexities in [25, Table 4]; Jaques and Schrottenloher analyse their "Local Prefix-based walk" and "Local Multi-Grover" algorithms under both the G-cost (passivey corrected quantum memory) and DW-cost (actively corrected quantum memory) metrics [24]. On the right "Best classical" side is a direct application of Equation (1) with $w \in \{2^{40}, 2^{64}, 2^{96}\}$. On the quantum side, the numbers are *"likely to underestimate the real cost by constant or poly-logarithmic factors"* [25]. On the classical side, the numbers ignore the cost of the isogeny oracle by setting $t = 1$ in (1). The lowest of the two (best quantum or best classical) complexities is in bold.

instance	Best quantum			Best classical		
	2^{96}	2^{64}	2^{40}	2^{96}	2^{64}	2^{40}
SIKEp434	124	147	178	**117**	**133**	**135**
SIKEp503	**134**	179	234	142	**158**	**160**
SIKEp610	**181**	189	307	183	**199**	**201**
SIKEp751	**219**	274	345	235	**251**	**253**

Table 1. The best known quantum and classical attack complexities (rounded base-2 logarithms) for the four SIKE instances. Further explanation in text.

Observe that the concrete cost on both sides is never less than $p^{1/4}$. Moreover, the only column under which the quantum attack has a (only slightly) better complexity is when the "MAXDEPTH" is 2^{96}. As discussed in the previous section, setting $w = 2^{80}$ on the classical side assumes more memory than currently exists on the planet. And, on the quantum side, NIST says that using MAXDEPTH 2^{96} is the *"approximate number of gates that atomic scale qubits with speed of light propagation times could perform in a millennium"* [47, p. 17].

Summary. I think it would be poetic to have a post-quantum standard where quantum computers don't really help to break it.

4 Concrete cryptanalytic clarity

"NIST believes it is important to understand how exactly this CoreSVP security translates into "true" bit security strength for KYBER."

– NIST [38]

One advantage the learning with errors (LWE) problem [42] has over the supersingular isogeny (SSI) problem is its accessibility. For a general practitioner, understanding a noisy matrix-vector product is much less daunting than understanding the elliptic curve group law, isomorphism classes, torsion points, isogeny kernels, expander graphs, and the list goes on. My guess is that there exist orders of magnitude more people that could sketch the general idea behind the LWE problem than there are people who could do the same for the SSI problem. I see this as a more serious obstacle for the adoption of SIKE than I do its time on the scene (see Section 1) or its present performance penalty (see Section 6). From time to time I have encountered numerous respected cryptographers who, despite decades of cryptanalytic evidence to the contrary, feel more at ease with the hardness of factorisation and finite field discrete logarithms than they do with that of the ECDLP, and I think this boils down to one reason: familiarity. Although not quantifiable with any scientific metric, in my view the role of familiarity in a cryptographic standardisation process cannot be overstated. From a constructive standpoint, there is no question that (R-)LWE-based protocols score a point over SIKE in the familiarity column.

When it comes to cryptanalysis, however, and in particular when viewing the lucidity of the best-known attacks against these protocols, the tables turn completely. There is unanimous consensus among isogenists that the best known attack against the SSI problem is vOW, and multiple analyses of its concrete runtime in the context of solving the SSI problem [1,24,15] confirm the applicability and precision of the original van Oorschot-Wiener analysis from 1999 [49]. Subsequently, Equation (1) takes as input a single SIKE parameter, the available memory, the number of parallel processors, and the time required to compute an isogeny, and outputs the expected time (or gates, or instructions, or any suitable metric you like [22, §5]) to find the secret isogeny and solve SSI. In conjunction with quantum algorithms not offering much help, having a closed formula for the concrete classical complexity makes it straightforward to get accurate estimates for the concrete complexity of the SSI problem in order to target a given security level. Moreover, the vOW algorithm is a *generic* collision finding algorithm; I can envision an hour-long talk being sufficient to describe the expander graphs underlying SSI and the nuts and bolts of the vOW algorithm such that an audience would then be up to speed with the state-of-the-art in SSI cryptanalysis.

The situation regarding the concrete complexity of the (R-)LWE problem is far from a closed formula, let alone one that has converged to a state of consensus among the experts. Lattice-based cryptanalysis is notoriously difficult to grasp with real competency[3], and there are only a handful of people with a comprehensive understanding of it. Moreover, the field is still very much in a state of flux, and this is not exactly the firmest ground to stand on when five of the seven NIST finalists are based on structured lattices (in addition to the security of one of the other two recently suffering a significant blow [9]). I think it would be an interesting experiment in community consensus to ask each of those five lattice-based submission teams to independently give a precise description of the best-known algorithm for one fixed instance of R-LWE or M-LWE, together with a concrete runtime analysis. Public key cryptographers have long been accustomed to consensus closed formulas for the runtime of the best known algorithms for mature problems like the ECDLP, so accompanying a proposed post-quantum standard by (at the very least) a "true" bit security of the proposed best classical attack seems like a low bar to set.

Summary. While understanding lattice-based cryptosystems is typically much easier than understanding the SIKE cryptosystem, understanding the state-of-the-art in attacking the SSI problem is much easier than understanding the state-of-the-art in attacking the LWE problem.

[3] This difficulty of gaining expertise in the cryptanalysis is unrelated to the actual computational difficulty of the problem. For example, I would also claim that gaining competency in braid group cryptanalysis is more difficult than gaining a firm understanding of vOW.

5 Side-channel security

> *"It is a very well-understood operation. We know how to attack it, we also know how to defend against it."*
>
> – David Jao [4].

Much of what follows in this section was said by Jao in his Round 3 seminar, but I will elaborate on some points just a little. There are two phases that take place during secret isogeny computations:

1. *Scalar multiplication.* Given the secret integer k and two public points P and Q on the public elliptic curve E, compute the secret (kernel) point $S = Q + [k]P$.
2. *Isogeny computation.* Given the secret point S and the public curve E, compute the image of E (and, during `keygen`, some public points) under the secret isogeny $\phi\colon E \to E/\langle S \rangle$.

The above quote is about the first phase, where we now have over two decades of research into protecting elliptic curve scalar multiplications. This first phase is where the secret key is used directly and where simple timing attacks are most relevant. The SIKE library uses a simple, constant-time Montgomery-style ladder to compute this scalar multiplication without any secret branching to protect against such attacks. An important point to note is that the first step is no more than 20% of the isogeny computation(s). Thus, even in a scenario where a 'bells-and-whistles' implementation throws all the compatible tricks in the book to guard against more sophisticated (e.g., power analysis) attacks, even a hypothetical 5x slowdown of this phase would not result in more than a 2x slowdown of the full isogeny computation.

At the conclusion of the first step, the secret scalar k is no longer used. The secret is no longer an integer but an elliptic curve point S, which in practice amounts to a projective representation of the x-coordinate of S, written as $(X_S\colon Z_S)$, where X_S and Z_S are both in $\mathbb{F}_{p^2}$. The order of the point S is ℓ^e, where $\ell = 2$ or $\ell = 3$ (depending on the side of the protocol). For the sake of concreteness, let's assume $\ell = 2$ so that $|\langle S \rangle| = 2^e$, but the discussion for $\ell = 3$ is analogous. The sequence of operations performed in the second phase is public, i.e. there is no data-dependent branching during the isogeny computation. Initially, the two field elements corresponding to S contain all of the e bits of secret entropy, and the curve E (whose single Montgomery coordinate is also carried through projectively via two field elements) is public and contains no secret entropy. The secret ℓ^e-isogeny is computed via a sequence $\phi_1, \ldots, \phi_e$ of e individual ℓ-isogenies. Each time an ℓ-isogeny is computed, it is evaluated at the secret torsion point(s) and the secret image curve; this essentially chews up a bit of entropy from the torsion point(s) and transfers it to the image curve. At the last step, all of the entropy is contained in E', the image of the secret isogeny $\phi\colon E \to E'$, and the kernel point S has (by definition) been moved to the zero point on E' under ϕ.

Assuming the first phase is adequately protected, attacking this second phase makes life more difficult for a side-channel adversary, whose goal is now to recover (partial) knowledge of these *field elements*. Here coordinate randomisation [12, §5.3] is rather straightforward; any $\lambda \in \mathbb{F}_p^\times$ can be used to randomise secret torsion points $(X\colon Z) \sim (\lambda X\colon \lambda Z)$ and secret image curves $(A\colon C) \sim (\lambda A\colon \lambda C)$ throughout the isogeny computation, in order to mitigate sophisticated power analysis attacks. If some special scenarios require it, it would be possible to randomise after each ℓ-isogeny computation with little impact on performance. An implementation that wants to be protected against specific types of attacks while minimising the overhead could make use of the above observations about entropy; initially, manipulation of (bits of) the torsion point coordinates are more difficult to guess than those of the curve coefficient, so randomisation of the latter is more important.

Though I'm personally dubious about the relevance of some of the fault attacks found in the literature, to me the most interesting side-channel observation to date is Ti's fault injection attack [48]. His observation that any perturbation in the auxiliary torsion points is likely to leak

[4] `https://www.nist.gov/video/implementation-isogeny-based-cryptography`.

large information about the secret is important if there are scenarios where such an attack would be relevant. However, it should be noted that Ti's simple countermeasure of checking the order of the auxiliary points prior to their publication [48, §3.3] mitigates this completely. Again, these order checks would only introduce a relatively small overhead on top of the isogeny computation and would be mandatory in scenarios where such perturbations are possible.

Summary. Decades of ECC side-channel analysis have given SIKE a good head start in the knowledge and implementation of side-channel protection, and protecting it in most scenarios would be relatively cheap.

6 The efficiency drawback

> *"The main drawback to SIKE is that its performance is roughly an order of magnitude worse than many of its competitors. Much work has been done to optimize implementations, including the compressed-key version, and it is hoped that such optimizations continue."*
>
> – NIST [38]

SIKE's relatively short time on the scene is often used as a security diss (cf. the quotes back on page 1), but on the other hand, it also means that SIKE is probably much further from its performance maturity.

Consider the performance evolution of the two fields most similar[5] to isogeny-based cryptography.

- *ECC.* Fifteen years after it was first proposed by Koblitz and Miller, ECC was standardised by NIST in FIPS 186-2 in the year 2000. The following year, Brown, Hankerson, López-Hernández and Menezes reported scalar multiplications on NIST Curve P-256 running on a Pentium II in just under 2 million cycles [10]. Five years later, Bernstein's 2006 implementations of Curve25519 scalar multiplications were more than twice as fast, running on a Pentium III in just over 800,000 cycles [6]. Fast forward a decade, and speed records for ECC scalar multiplications targeting the 128-bit security level require less than 60,000 cycles on an Intel Core i7 [13, Table 3].
- *PBC.* In his 2000 paper introducing one of the first pairing-based protocols, Joux reported that the Tate pairing could be computed in one second on a Pentium II processor [26]. A decade later, Beuchat, González-Díaz, Mitsunari, Okamoto, Rodríguez-Henríquez and Teruya broke the millisecond barrier for the first time (for a pairing targeting 128-bit security) on an Intel Core i7 [8]. In the following year, Aranha, Karabina, Longa, Gebotys and López-Hernández [4] gave timings close to half a millisecond at the same security level.

Both fields started out with the reputation of being relatively slow. ECC started out with small keys as its selling point, but RSA was more efficient across the board. It did not take too long for ECC to outperform RSA at key generation and signing, but in 2012 RSA-2048 verified signatures in 98,000 Sandy Bridge cycles, roughly 13 times faster than ECDSA verifications on NIST Curve P-224 [20, §1]. Almost a decade later and Ed25519 verifications have closed the gap to within a factor of two of RSA-3072, while Schnorr$\mathbb{Q}$ verification speeds have overtaken it[6].

The story for pairing computation is similar; pairings were seen as prohibitively slow for real-world deployment in the early days, but a decade of intense research saw speeds improve more than three orders of magnitude over Joux's initial timing.

[5] Similar in terms of the types of operations involved, not the security story. Pairings have arguably had a bumpy security story in the last decade (though this is not the fault of the elliptic curve groups!).

[6] For the (`keygen`, `sign`, `verify`) operations on a Skylake processor, `https://bench.cr.yp.to/results-sign.html` reports (823177475?,8722834,82706) cycles for RSA-3072 and (45719,48796,165025) cycles for Ed25519; Longa reported to me that the numbers (on a Skylake processor with turboboost disabled, etc) for Schnorr$\mathbb{Q}$ are (40897,34463,62318).

The point is that while curve-based cryptosystems never start out blazingly fast, their structure offers opportunities for tricks and speedups that has, historically speaking, brought them to a point where speed becomes either a non-issue, or better yet, a selling point. In the case of isogeny-based cryptography, I am optimistically anticipating a similar trend, while at the same time acknowledging that a scheme cannot be standardised for merely showing the potential to get faster (if its relative inefficiency makes it a deal-breaker in practice). However, in echoing what Jao said in his recent talk 4, the relative rate of improvement of isogeny computations has been greater than those of its code- and lattice-based counterparts of late, and I would add that there is no reason to believe this trend will stop anytime soon. In the past decade, isogeny computations have been accelerated from 245ms @2.4GHz in 2011 to 5.9ms @3.4GHz today. Even in the unlikely scenario where there are no further algorithmic breakthroughs in isogeny computation[7], it is inevitable that processors will continue to get faster and include new instruction sets that are welcomed by isogeny implementers.

Summary. When it comes to curve-based cryptography versus its counterparts, performance disparity tends only to be temporary[8].

7 Happy hybrids

> *"The submission package shall include a statement that lists and describes the advantages and limitations of the cryptosystem. [...] This could include, for example, the suitability of the algorithm for use in hybrid schemes..."*
>
> – NIST [47, §2.B.6]

Although it makes perfect sense to focus solely on the post-quantum algorithms, in SIKE's case it is definitely a bummer that NIST's call for proposals deemed hybrid modes out of scope [47, §1]. All of the "transition strategy" talks and panels I've listened to suggest that hybrids will be enabled by default until some of the standardisation dust settles, and as far as I can tell SIKE is the only scheme that offers a slick hybrid with minimal (speed, implementation, and codebase) overhead.

The idea [14] is to exploit the base field arithmetic and Montgomery ladder implementations from the SIKE implementation to do X25519- and X448-style key exchange, the modern standard in ECDH [29]. For example, the arithmetic in SIKEp434 takes place in $\mathbb{F}_{p^2}$ with the 434-bit prime $p = 2^{216}3^{137} - 1$. Applying the deterministic generation procedure in [29, App. A] over the base field $\mathbb{F}_p$ produces the Montgomery curve $\mathcal{E}/\mathbb{F}_p\colon y^2 = x^3 + \mathcal{A}x^2 + x$ with $\mathcal{A} = 439322$. The curve $\mathcal{E}$ has order $\#\mathcal{E} = 4r$ and its quadratic twist $\mathcal{E}'$ has order $\#\mathcal{E}' = 4r'$, where r and r' are 432-bit primes. Following the modern standards [29], an ECDH public key would be one element of $\mathbb{F}_p$; this would increase the 330-byte uncompressed public keys of SIKEp434 by 55 bytes (17%) and add an overhead of less than 15% [14]. The size increase to the 197-byte compressed public keys is 28%, but the relative performance overhead would then be even less than the uncompressed case.

Haters might say that such big ECDH security is classical overkill that adds unnecessary bytes to the public keys and ciphertexts. To them I would respond by (1) pointing out the NSA's recent move[9] from 256-bit ECC in Suite B to 384-bit ECC in the CNSA suite, (2) pointing out that a huge chunk of the work in maintaining a high-performance ECC or isogeny-based library goes into the tailored assembly-optimised field arithmetic, and (3) telling them to worry about their own post-quantum key sizes!

There is a neat possibility that the above hybrid solution allows out of the box, in that the generator for the ECDH part of the protocol can be fresh for every public key at no additional cost. For example, Alice could complete the generation of her SIKE public key, and then define her

[7] Though not directly applicable to SIKE, a huge breakthrough came in 2020 [7].

[8] ... but keysize disparity lasts forever!

[9] `https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm`.

ECDH generator (corresponding to that public key) using one of the $\mathbb{F}_p$ elements in it. Suppose $u \in \mathbb{F}_p$ is this element, then setting $u = x(P)$, i.e. the x-coordinate corresponding to $P \in \mathcal{E}$ or $P \in \mathcal{E}'$, her and Bob can safely take the generator for their ECDH instance as $x(Q)$ with $Q = [4]P$, for which they are guaranteed that $Q \in \mathcal{E}[r]$ or $Q \in \mathcal{E}[r']$, i.e. that the generator Q is in one of the large prime order subgroups[10]. Random self-reducibility of discrete logarithms tells us that no generator is better than another for a single ECDLP instance, but a fresh generator for each public key does provide protection against the Silverman-Stapleton batch discrete logarithm algorithm (see [28]) or against any hypothetical/unknown *all-for-the-price-of-one*-style attacks that might make use of a fixed generator. This comes at no additional performance or bandwidth cost.

Summary. SIKE is the only NIST candidate that has a nice hybrid.

8 Other avenues of attack

> *"We can therefore conclude that at least heuristically, it seems extremely unlikely that Petit's attack can possibly apply to the actual, balanced SIDH parameters. "*
>
> – Chloe Martindale and Lorenz Panny [33, §4.3]

The most impactful SIDH cryptanalysis to date is the 2016 active attack due to Galbraith, Petit, Shani and Ti [19]. They showed that using a static public key opens the protocol up to an active adversary that can fully recover the corresponding secret key in as few interactions as the bitlength of the secret. It is important to stress that this is not an attack on the SSI problem, but rather on the deployment of it in a key exchange protocol that does not validate static keys. Fortunately, all of the competitors in the NIST PQC standardisation process are on a level playing field in this regard: codes and lattices also fall prey to active attacks, which is why all such schemes use some transformation (a la Fujisaka-Okamato [18]) to morph them into a secure key encapsulation mechanism (KEM).

Another important line of cryptanalytic works are the algorithms that exploit the torsion point images and known endomorphism rings as pioneered by Petit [39]. Recall that the degrees of Alice and Bob's secret isogenies in the SSI problem [23, Problem 5.1] are $A \approx B \approx \sqrt{p}$. If the secret isogenies are instead *unbalanced* such that $B > A^4 > p^4$, Petit shows [39, Proposition 2] that the additional torsion point information can be used to recover one of the secret isogenies. Since his initial paper, a number of papers have given improvements to the attack, and these have helped shape the wider security landscape of isogeny-based cryptography; in particular, they warn against problem and protocol variants for which the secret isogenies are unbalanced. In terms of SIKE, however, it must be emphasised that (at present) these attacks do not give improvements to the generic collision- or claw-finding attacks for the original SSI problem.

It must also be stressed that if torsion point attacks are improved to the point where they do become relevant for SIKE, the protocol can be tweaked so the endomorphism ring of the starting curve is kept hidden to the adversary: during static key generation, Alice can privately walk to a new starting curve and simply include its description (i.e. A-coefficient or j-invariant) as part of her public key. The rest of the public parameters are points on the starting curve which can be deterministically and efficiently obtained (by both Alice and Bob) using methods from the line of works on public key compression [5]. This would add a small overhead to the performance and a moderate inflation to Alice's public key (less than 35% for uncompressed keys and around 50% for compressed keys), but nowhere near enough to threaten SIKE's place as the leader of the pack in the bandwidth category. To my knowledge, none of the attacks in this line of work remain applicable if the endomorphism rings of both curves in the statement of the SSI problem are hidden.

A nice paper by Martindale and Panny [33] (quoted above) details a range of attack avenues they tried in attempts to exploit the structure that underlies the SSI problem, including the possibility of applying Petit's attack on the balanced SIKE parameters. Though their categorisation

[10] The other option is to fix two zeros in the secret keys and use P as is à la X25519.

of the attack attempts as *negative results* is not meant as a definitive one, for now it does serve as a positive in SIKE's favour; it is evidence that smart people have been working to try and break SSI, but ultimately that the best attacks against SIKE remain generic collision-finding attacks against the SSI problem.

On a related note, it is also worth reiterating that, based on current knowledge, solving the (computational and decisional) Diffie-Hellman problems appearing in the context of SIKE is no easier than solving the SSI problem directly.

Summary. A lot of great cryptanalytic work has improved our knowledge of the isogeny problem landscape in the last decade, but the best attack against the SSI problem and against SIKE remain the generic collision-finding attacks.

9 Elegance

> *"It is my intent to show that elliptic curves have a rich enough arithmetic structure so that they will provide a fertile ground for planting the seeds of cryptography."*
>
> – Victor S. Miller [37]

Even after more than a decade of working on ECC, I am still occasionally taken aback by its most foundational building block: the elliptic curve group law. The fact that a geometric object (i.e. a curve) comes hand-in-hand with an entirely algebraic description (i.e. a group law) is a pleasant surprise time and time again. But the great thing about the story of elliptic curves in cryptography is that this is just the beginning of the surprises; the seeds that Miller and Koblitz planted three and a half decades ago have since blossomed and propagated in the most striking ways, and supersingular isogeny graphs are a gorgeous case in point.

Around the same time ECC was introduced, the study of supersingular isogeny graphs had concurrently been initiated by Mestre [36]. Though this line of inquiry was not undertaken with cryptographic applications in mind, by the time Shor's algorithm showed up to start spoiling the public key party in 1994 [44], Pizer [40] had shown that supersingular ℓ-isogeny graphs were $(\ell+1)$-regular Ramanujan graphs and set the stage for what would come almost two decades later: Jao and De Feo's answer to the whole quantum conundrum.

It is an absurdly cool fact that *every* prime $p > 3$ precisely defines a set of close to $p/12$ elements, for which *every* other prime ℓ induces an $(\ell + 1)$-regular Ramanujan graph. Even for moderate values of p (far below what is used in practice), storing an expander graph on a computer is impossible; by the time we ramp up to the graph underlying SIKEp434, the number of nodes we need to store (to say nothing of the edges) is more than the number of atoms in the observable universe. The rapid expansion property of these graphs makes the situation chaotic for an adversary who would like to navigate around in any meaningful way. Intuitively, expander graphs feel like the perfect place to instantiate public key cryptography. They are part of the reason why, to me, Jao and De Feo's solution stands alone among the post-quantum key encapsulation schemes in terms of its elegance.

Some isogenists I've spoken to seem to instinctually feel that SIDH would be more *pure* if not for the auxiliary torsion points. While conceding that their presence allows for the active GPST attack in the previous section, after 10 years on the scene and a stable classical and quantum attack story, I only see the torsion points as a success. After all, they are what Jao and De Feo used to help Alice and Bob magically commute in the expander graph, when (over on the other side of the Deuring correspondence) the non-commutative endomorphism rings suggest they should not be able to. And, importantly, this non-commutativity is what has protected SIKE from any sort of quantum attacks that pose more of a threat than generic vOW.

Of course, in a competition to decide which cryptography is going to end up being used in practice, there should be no points for style or beauty. The choice of which scheme is best should be that which is most likely to offer the strongest protection to the widest base of users. But there

is one practical benefit that SIKE's elegance affords it over all of its remaining competitors: SIKE is *perfectly correct*, i.e. the absence of dirty words like noise and error means there is never any worries about reconciliation or decryption failures. The Crypto ePrint archive contains at least 10 papers that have been written on analysing and attacking imperfect schemes (like Kyber and Saber) since NIST acknowledged the importance of these failures in their original call [47, §4.B.4]. The point is that the comparative elegance of SIKE is not just an aesthetic feature; in regard to decryption failures, it actually reduces the attack surface.

Summary. In terms of the remaining key encapsulation candidates, a search over each of the Round 3 specification documents for the number of instances of the string "failure" finds (in alphabetical order) BIKE = 27, FrodoKEM = 29, HQC = 16, Kyber = 61, McEliece = 18, NTRU = 6, NTRU Prime = 42, Saber = 24, and finally, SIKE = 0.

10 The SIKE challenges

"When it comes to betting on yourself [...] you're a chicken-livered coward if you hesitate."

– Bertie Charles Forbes [16, p. 195]

At NIST's 3rd standardisation conference in June of this year, two cash bounties were announced for the solutions of the supersingular computational Diffie-Hellman (SSCDH) problem on two small SIKE instances:

- $5,000 USD is on offer for the solution of the SSCDH problem on `$IKEp182`, a mini instance with $p = 2^{91} \cdot 3^{57} - 1$.
- $50,000 USD is on offer for the solution of the SSCDH problem on `$IKEp217`, a small instance with $p = 2^{110} \cdot 3^{67} - 1$.

These challenge instances, together with the code used to generate them, can be found at

`https://github.com/microsoft/SIKE-challenges`.

The larger of these two instances, `$IKEp217`, is defined over a prime of exactly half the bitlength of that which underlies `SIKEp434`, the smallest parameter set in the SIKE submission. According to the current SIKE security analysis, this places `$IKEp217` significantly below 64 bits of classical security. These instances were chosen so that a relevant classical cryptanalytic breakthrough should put them well within reach, but failing that, a significant (but feasible) amount of computational resources will be required to mount a meet-in-the-middle or vOW attack.

I conclude by encouraging the proponents of rival schemes to also publish smaller challenge instances corresponding to their cryptosystems[11], independently of whether or not they put any money where their mouths are.

Summary. If you are not yet convinced that breaking SIKE is hard, then perhaps some prize money will encourage you to get *cracking*.

[11] Challenges for hard lattice problems have been around for some time, see `https://www.latticechallenge.org/` and `https://web.eecs.umich.edu/~cpeikert/rlwe-challenges/`. However, their connection to the related schemes under consideration in the NIST competition is unclear (at least to me). What I am encouraging is that the teams propose concrete toy challenges under the same security analysis that their real-world parameters are derived; e.g., publishing a Kyber instance with 2^{64}-bit security.

References

1. G. Adj, D. Cervantes-Vázquez, J. Chi-Domínguez, A. Menezes, and F. Rodríguez-Henríquez. On the cost of computing isogenies between supersingular elliptic curves. In *SAC 2018*, pages 322–343, 2018.
2. L. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *SFCS 1979*, pages 55–60, 1979.
3. D. Apon. Passing the final checkpoint! NIST PQC 3rd round begins. August, 2020. `https://www.scribd.com/document/474476570/PQC-Overview-Aug-2020-NIST`.
4. D. F. Aranha, K. Karabina, P. Longa, C. H. Gebotys, and J. C. López-Hernández. Faster explicit formulas for computing pairings over ordinary curves. In *EUROCRYPT 2011*, pages 48–68, 2011.
5. R. Azarderakhsh, D. Jao, K. Kalach, B. Koziel, and C. Leonardi. Key compression for isogeny-based cryptosystems. In *ASIAPKC*, pages 1–10, 2016.
6. D. J. Bernstein. Curve25519: New Diffie-Hellman speed records. In *PKC 2006*, pages 207–228, 2006.
7. D. J. Bernstein, L. De Feo, A. Leroux, and B. Smith. Faster computation of isogenies of large prime degree. *Open Book Series*, 4(1):39–55, 2020.
8. J. Beuchat, J. Enrique González-Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya. High-speed software implementation of the optimal ate pairing over Barreto-Naehrig curves. In *Pairing 2010*, pages 21–39, 2010.
9. W. Beullens. Improved cryptanalysis of UOV and Rainbow. To appear at EUROCRYPT 2021.
10. M. Brown, D. Hankerson, J. C. López-Hernández, and A. Menezes. Software implementation of the NIST elliptic curves over prime fields. In *CT-RSA 2001*, pages 250–265, 2001.
11. D. Coppersmith and A. Shamir. Lattice attacks on NTRU. In *CRYPTO 1997*, pages 52–61, 1997.
12. J. Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In *CHES 1999*, pages 292–302, 1999.
13. C. Costello and P. Longa. Four$\mathbb{Q}$: Four-dimensional decompositions on a $\mathbb{Q}$-curve over the Mersenne prime. In *ASIACRYPT 2015*, pages 214–235, 2015.
14. C. Costello, P. Longa, and M. Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In *CRYPTO 2016*, pages 572–601, 2016.
15. C. Costello, P. Longa, M. Naehrig, J. Renes, and F. Virdia. Improved classical cryptanalysis of SIKE in practice. In *PKC 2020*, pages 505–534. Springer, 2020.
16. B. C. Forbes. *Keys to success, personal efficiency.* 1917.
17. G. Frey and H. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 62(206):865–874, 1994.
18. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO 1999*, pages 537–554, 1999.
19. S. D. Galbraith, C. Petit, B. Shani, and Y. B. Ti. On the security of supersingular isogeny cryptosystems. In *ASIACRYPT 2016*, pages 63–91, 2016.
20. M. Hamburg. Fast and compact elliptic-curve cryptography. Cryptology ePrint Archive, Report 2012/309, 2012.
21. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring-based public key cryptosystem. In *ANTS 1998*, pages 267–288, 1998.
22. D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. De Feo, B. Hess, A. Jalali, B. Koziel, B. LaMacchia, P. Longa, M. Naehrig, J. Renes, V. Soukharev, and D. Urbanik. SIKE: Supersingular Isogeny Key Encapsulation. Manuscript available at `sike.org/`, 2017.
23. D. Jao and L. De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto 2011*, pages 19–34, 2011.
24. S. Jaques and J. M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In *CRYPTO 2019*, pages 32–61, 2019.
25. S. Jaques and A. Schrottenloher. Low-gate quantum golden collision finding. In *SAC 2020, to appear. https://eprint.iacr.org/2020/424*, 2020.
26. A. Joux. A one round protocol for tripartite Diffie-Hellman. In *ANTS 2000*, pages 385–394, 2000.
27. T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, and D. Osvik. Factorization of a 768-bit RSA modulus. In *CRYPTO 2010*, pages 333–350, 2010.
28. F. Kuhn and R. Struik. Random walks revisited: Extensions of Pollard's rho algorithm for computing multiple discrete logarithms. In *SAC 2001*, pages 212–229, 2001.
29. A. Langley, M. Hamburg, and S. Turner. Elliptic curves for security. Internet Research Task Force RFC7748, 2016. `https://tools.ietf.org/html/rfc7748`.

30. P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 275–280, 1988.
31. J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, 1988.
32. P. Longa, W. Wang, and J. Szefer. The cost to break SIKE: A comparative hardware-based analysis with AES and SHA-3. Cryptology ePrint Archive, Report 2020/1457, 2020.
33. C. Martindale and L. Panny. How to not break SIDH. *CFAIL 2019*, 2019.
34. R. J. McEliece. A public-key cryptosystem based on algebraic codes. *Coding Thv*, 4244:114–116, 1978.
35. A. J. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.
36. J.F. Mestre. La méthode des graphes. exemples et applications. In *Proceedings of the international conference on class numbers and fundamental units of algebraic number fields (Katata)*, pages 217–242, 1986.
37. V. S. Miller. Use of elliptic curves in cryptography. In *CRYPTO 1985*, pages 417–426, 1985.
38. National Institute of Standards and Technology (NIST). Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. NISTIR 8309, 2020. `https://doi.org/10.6028/NIST.IR.8309`.
39. C. Petit. Faster algorithms for isogeny problems using torsion point images. In *ASIACRYPT 2017*, pages 330–353, 2017.
40. A. K. Pizer. Ramanujan graphs and Hecke operators. *Bull. of the AMS*, 23(1):127–137, 1990.
41. C. Pomerance. A tale of two sieves. In *Notices Amer. Math. Soc*, 1996.
42. O. Regev. The learning with errors problem (invited survey). In *CCC 2010*, pages 191–204, 2010.
43. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
44. P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS*, 1994.
45. J. Stern. A method for finding codewords of small weight. In *International Colloquium on Coding Theory and Applications*, pages 106–113, 1988.
46. S. Tani. Claw finding algorithms using quantum walk. *Theor. Comput. Sci.*, 410(50):5285–5297, 2009.
47. The National Institute of Standards and Technology (NIST). Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, December, 2016.
48. Y. B. Ti. Fault attack on supersingular isogeny cryptosystems. In *PQCrypto 2017*, pages 107–122. Springer, 2017.
49. P. C. van Oorschot and M. J. Wiener. Parallel collision search with cryptanalytic applications. *J. Cryptology*, 12(1):1–28, 1999.
50. J. van Tilburg. On the McEliece public-key cryptosystem. In *CRYPTO 1988*, pages 119–131, 1988.