Quantum circuits and approaches to parallelism for solving the CSSI problem

Reza Azarderakhsh[1], Jean-Françoise Biasse[2], Rami El Khatib[1], Brandon Langenberg[1], and Benjamin Pring[2]

[1] Department of Computer Science and Engineering, Florida Atlantic University
razarderakhsh@fau.edu, relkhatib2015@fau.edu, blangenb@fau.edu
[2] Department of Mathematics and Statistics, University of South Florida,
biasse@usf.edu, benjamin.pring@gmail.com

Abstract. In this paper, we improve upon the parallelisation of the Grover-based quantum claw-finding attack on the *Computational Supersingular Isogeny* (CSSI) problem studied in [15] and optimised in [3]. The CSSI problem is the underlying hard problem behind the SIKE cryptosystem [2]. We leverage specifics of the claw-finding problem, exploiting classical computation to surpass the limits on the performance of parallelisation of Grover proved by Zalka [24] under the assumption that the quantum oracle is a black box.
Our parallel attack improves on the previous attacks against SIKE [15] under constraints such as the MAXDEPTH (maximum quantum circuit-depth) and is particularly effective with respect to the MAXWIDTH (maximum memory) constraint which recently motivated the upgrade of the security level of SIKE p751 from NIST Level III to NIST level V [2].

Keywords: quantum cryptanalysis, quantum search, CSSI, SIKE

1 Introduction

Let $\chi : \{0,1\}^n \longrightarrow \{0,1\}$ be a boolean function such that $|\chi^{-1}(1)| = 1$. Grover's [11] algorithm finds the unique $x_* \in \{0,1\}^n$ such that $\chi(x_*) = 1$ with high probability in $O\big(2^{n/2}\big)$ calls to a quantum circuit implementing χ (the quantum oracle). Zalka [24] proved that Grover's quantum search offers poor parallelism compared to naive classical exhaustive search (which offers almost perfect parallelism) when the quantum oracle is treated as a black-box. Indeed, if we possess 2^s quantum computers, whilst the *individual* quantum circuit-size and circuit-depth may be reduced by a factor of $O(2^{-s/2})$, the *total* number of calls to the oracle over all 2^s quantum computers is in $O\big(2^{s/2}2^{n/2}\big)$.

Grover's algorithm is explicitly stated as one of the leading methods of quantum cryptanalysis in the SIKE key encapsulation method submitted to the NIST standardisation of quantum-resistant public-key cryptosystems [20]. This scheme

is based upon the hardness of the computational supersingular isogeny (CSSI) problem (see Definition 1), a problem whose origins lie in the work of [7]. The impact of the Grover-based attack on specific instances of SIKE (parametrized by a prime p) can be assessed via the estimation of the cost of the quantum oracle for the isogeny search. A comparison of this attack with the performance of Tani's algorithm [22] (which can attack SIKE with a query complexity of $O(p^{1/6})$) was studied by Jaques and Schanks [15], who showed that under realistic assumptions concerning the underlying quantum data structure required to implement Tani's algorithm, both Grover's algorithm and Tani's algorithm have a cost of $O(p^{1/4})$, raising the question of which may offer better performance.

These two assumptions — the *overhead* of the quantum oracle and the *parallelism scaling* of Grover's algorithm are intrinsic to the NIST call for proposals [21]. The authors therefore believe that the study of these assumptions are of importance, as they are a cultural assumption. In this paper, we show how to improve the parallelism of the Grover-based technique, extending the work of [3] and estimate the full cost of the attack for instances of SIKE whose security is defined the size of the prime p for instances where $\log(p) = 434, 503, 610, 751$.

Contributions In this paper we make the following contributions

- We describe a parallelisation of Grover's algorithm over 2^s processors which offers a total cost of $O(2^{s/4}2^{p/4}C)$ which should be compared to a black-box parallelism, which offers $O(2^{s/2}2^{p/4}C)$, where C contains polynomial factors.
- We show our method offers improvements over the state of the art under MAXWIDTH=2^{96} and MAXDEPTH=2^{96} against the instances of SIKE defined by $\log(p) = 434$ and 751.
- We estimate the resources to implement the attack, including the quantum circuit to compute a degree-2^e isogeny path based upon methods of [14].
- We comment upon assumptions and cost models in quantum cryptanalysis.

2 Background

SIKE and the CSSI problem The Computational Supersingular Isogeny problem is the underlying hard problem behind the SIKE key encapsulation mechanism [2]. The security of SIKE instantiations are parameterised by a prime of the form $p = 2^e3^f - 1$ such that $2^e \approx 3^f$ and an elliptic curve E_1 defined over $\mathbb{F}_{p^2}$, whilst the key-exchange transmission reveals a second elliptic curve curve E_2 also defined over $\mathbb{F}_{p^2}$ and guarantees that there exists an isogeny (a morphism) $\phi : E_1 \longrightarrow E_2$ with a kernel of size 2^e (the degree of the isogeny). Finding this degree-2^e isogeny $\phi : E_1 \longrightarrow E_2$ is equivalent to breaking SIKE. A similar hardness assumption was previously introduced in [7] by Charles, Goren and

Lauter without restriction on the degree of the isogeny. See [8] for its connection with other hardness assumptions [3]

Definition 1 (CSSI problem [14]). *Let E_1, E_2 be two supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ such that there is a unique degree 2^e isogeny $\phi : E_1 \longrightarrow E_2$ (up to isomorphism) with $e \approx \frac{\log_2 p}{2}$. Given E_1, E_2, p and e, the Computational SuperSingular Isogeny (CSSI) problem is to find the unique isogeny between E_1 and E_2.*

The problem implictly defines a *Ramanujan graph* where each node is an elliptic curve and edges are degree 2 isogenies between these curves. For any two elliptic curves E' and E'' we have that $E' \cong E''$ if and only if $j(E') = j(E'')$, where $j(\cdot)$ is the j-invariant (which is efficiently computable) of a given elliptic curve — this allows us to assign each node in this graph a unique label. Note that in a SIKE instance, there is a negligible probability that more than one isogeny exists, while the CSSI problem offers the guarantee that the solution is unique.

Given the graph-based interpretation of this problem as discussed above, one approach to solving this problem via both classical and quantum methods is the *meet-in-the-middle* approach to *claw-finding*. In this scenario we allow $e = e_1 + e_2$ and attempt to find a node that corresponds to a degree-2^{e_1} isogeny starting from E_1 and a degree-2^{e_2} isogeny starting from E_2. If we can find such a node (an elliptic curve identified by its j-invariant) then we can use these isogenies to generate an isogeny of degree $2^{e_1+e_2} = 2^e$ from E_1 to E_2. Both classical [1, 9] and quantum [15] approaches to this attack methodology have been studied.

Classically, one can either enumerate and sort a table of j-invariants or use the van Oorschot-Weiner [23] (VW) parallel-collision finding approach to find the unique claw we are searching for. Asymptotically (modulo polynomial factors) these approaches require $O(p^{1/4})$ classical gates given unbounded memory.

In terms of quantum attacks, the best theorised approaches are either Grover's algorithm [11] or Tani's algorithm [22] — cost estimates for these algorithms under realistic assumptions concerning quantum memory and error-correction were studied in [15]. In particular, the results of [15] reduce the asymptotic complexity of solving the CSSI problem via Tani's algorithm from $O(p^{1/6})$ to $O(p^{1/4})$ — giving it an identical asymptotic complexity (modulo polynomial factors) to that of using Grover's algorithm to solve the CSSI problem.

Quantum search techniques An n-qubit (quantum bit) quantum state can be expressed relative to the computational basis as $\left\{ |x\rangle : x \in \{0,1\}^n \right\}$ by $|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$, where $\alpha_x \in \mathbb{C}$ and $\sum_x |\alpha_x^2| = 1$. Crucially, we can perform a *measurement* of this state which collapses $|\psi\rangle$ into a the classical bitstring $x \in \{0,1\}^n$ with probability $|\alpha_x^2|$. Quantum states can be regarded as *state*

[3] This section and details on SIKE can be expanded upon if the reviewers wish, but for the 12-page submission the authors believe that this is sufficient information.

vectors so that $|\psi\rangle \in \mathbb{C}^{2^n}$. Given this interpretation, the space of all possible quantum algorithms (which do not include measurement) acting upon n-qubits is the set $\left\{U \in \mathbb{C}^{2^n \times 2^n} : UU^\dagger = U^\dagger U = I\right\}$ of *unitary operators*, where $\dagger$ is the conjugate-transpose operator.

A simple quantum algorithm can therefore be thought of as a sequence of unitary operations which increase the magnitude of α_x (which encodes information we wish to learn) followed by a measurement, which gives us a high probability of obtaining this information. *Quantum amplitude amplification* [5], a generalisation of *Grover's algorithm* [11] allows us to take a quantum algorithm that results in useful information with probability a and increase this probability close to 1. Crucially, quantum amplitude amplification gives an asymptotic advantage in obtaining this information — if we simply repeated the initial quantum algorithm we would require $O(\frac{1}{a})$ applications, measurements, whereas with quantum amplitude amplification we would only require $O(\frac{1}{\sqrt{a}})$ such applications, but require a *quantum oracle*, which is simply a quantum circuit that recognises the elements of $x \in \{0,1\}^n$ we are interested in. Quantum circuits are themselves constructed out of primitive *quantum gates* — we fix the choice of the Clifford+T [19] universal quantum gate set comprised of the Clifford gate set and the T-gate, which is sufficient to exactly the quantum circuits we discuss.

Definition 2 (Success probability of a quantum algorithm). *Let $\mathcal{A}$ be any quantum algorithm acting upon n-qubits and $\chi : \{0,1\}^n \longrightarrow \{0,1\}$. We say that the success probability of $\mathcal{A}$ relative to χ is the probability that measuring the state $\mathcal{A}|0^n\rangle$ in the computational basis results in $x \in \{0,1\}^n$ such that $\chi(x) = 1$.*

Note that $\mathcal{A}$ can have different success probabilities relative to different boolean functions — a trivial example is the two constant boolean functions on n-bits.

Definition 3 (Quantum oracle). *The quantum oracle $\mathcal{O}_\chi$ defined by the boolean function $\chi : \{0,1\}^n \longrightarrow \{0,1\}$ has the following action upon the set of n-qubit computational basis states $\left\{|x\rangle : x \in \{0,1\}^n\right\}$*

$$\mathcal{O}_\chi |x\rangle \begin{cases} -|x\rangle & \text{if } \chi(x) = 1 \\ |x\rangle & \text{otherwise.} \end{cases} \tag{1}$$

We note as a fact that quantum oracles can be implemented via the set of quantum gates $\{X, \wedge_1(X), \wedge_2(X)\}$ which respectively implement *reversible* versions of the classical operations $\neg, \oplus$ and $\wedge$, which are sufficient to implement any boolean function. These gates must be reversible owing the aforementioned unitary property that gives us that any quantum algorithm excluding measurement must possess an inverse. The $\wedge_k(X)$ gate is a generalisation, acting on $k+1$ computational basis states by $\wedge_k(X)|x_1 \ldots x_k\rangle|x_{k+1}\rangle \mapsto |x_1 \ldots x_k\rangle|x_{k+1} \oplus x_1 \cdots x_k\rangle$ so that $X = \wedge_0(X)$ and $\wedge_0(X)|x_1\rangle \mapsto |x_1 \oplus 1\rangle$. These gates can be exactly synthesised using the Clifford+T *universal quantum gate set* [19] and we use the costs for the $\wedge_k(X)$ from [18]. We denote the cost of executing an arbitrary

quantum operation $\mathcal{A}$ by $C_{\mathcal{A}}$. As all operations in the statement of amplitude amplification are serial (see $Q(\mathcal{A}, \mathcal{O}_\chi, k)$ in Theorem 1), we can substitute either quantum circuit-size or quantum circuit-depth of $\mathcal{A}$ to obtain the relevant cost.

Theorem 1 (Amplitude amplification — [5]). *Let $\mathcal{A}$ be any quantum algorithm (with inverse $\mathcal{A}^\dagger$) with a probability of success relative to $\chi : \{0,1\}^n \longrightarrow \{0,1\}$ of $a \in [0,1]$. Then given $\mathcal{A}, \mathcal{O}_\chi$ and k, there exists a quantum algorithm $Q(\mathcal{A}, \mathcal{O}_\chi, k) := \left(\mathcal{A}\mathcal{O}_{\bar{n}}\mathcal{A}^\dagger\mathcal{O}_\chi\right)^k\mathcal{A}$, denoted by $\mathcal{B}(k)$ when there is no ambiguity, with a probability of success relative to $\chi : \{0,1\}^n \longrightarrow \{0,1\}$ of*

$$b(k) = \sin^2\Big(\big(2k+1\big)\cdot \arcsin\sqrt{a}\Big) \tag{2}$$

and which costs (where we assume $C_{\mathcal{A}} = C_{\mathcal{A}^\dagger}$)

$$C_{\mathcal{B}(k)} = k\cdot(C_{\mathcal{O}_\chi} + C_{\mathcal{O}_{\bar{n}}}) + (2k+1)\cdot C_{\mathcal{A}} \tag{3}$$

where $\bar{n} : \{0,1\}^n \longrightarrow \{0,1\}$ is defined by $\bar{n}(x) = 1$ iff $x \neq 1$.

Grover's algorithm is a simple application of quantum amplitude amplification which leverages the quantum algorithm $\mathcal{A} = H^{\otimes n}$ (the *Hadamard transform* on n-qubits). $H^{\otimes n}$ acts as $H^{\otimes n}\left|0^n\right\rangle \mapsto \frac{1}{2^{n/2}}\sum\limits_{x\in\{0,1\}^n}\left|x\right\rangle$, which creates the *uniform superposition*. The probability of obtaining an element such that $\chi(x) = 1$ is $a = \frac{|\chi^{-1}(1)|}{2^n}$, and Lemma 1 [4] (see Appendix B) shows that when $k = \left\lfloor \frac{\pi}{4\cdot\arcsin\sqrt{\frac{|\chi^{-1}(1)|}{2^n}}} \right\rfloor \leq \frac{\pi}{4}\cdot\sqrt{\frac{2^n}{|\chi^{-1}(1)|}}$, the probability of obtaining an element $x \in \{0,1\}^n$ such that $\chi(x) = 1$ is $\max\left\{1 - \frac{|\chi^{-1}(1)|}{2^n}, \frac{|\chi^{-1}(1)|}{2^n}\right\}$.

3 The cost of the isogeny oracle

To implement the claw-finding methodology in Section 2 with Grover's algorithm, we must construct a quantum circuit that accepts a secret $x \in \{0,1\}^{e_1}$ (which corresponds to our search-space), a classically known elliptic curve E_1 defined over $\mathbb{F}_{p^2}$, and two classically known torsion points $P, Q \in E_1(\mathbb{F}_{p^2})$. This circuit, denoted by $\mathcal{E}_{f_{e_1}}$, 1) computes an initial point $R = P + [x]Q$, then 2) computes the end-curve E' of a degree-2^{e_1} isogeny $\phi : E_1 \longrightarrow E'$ that is uniquely specified by R and finally 3) outputs the j-invariant of E'. This process is in fact part of the SIKE key-exchange mechanism and a detailed breakdown of the procedure based upon the $O\big(e\log_2 e\big)$ cost algorithm of [14] in terms of classical $\mathbb{F}_{p^2}$ arithmetic (based upon a projective coordinate representation of elliptic curve points) is provided in the SIKE specification [2]. Note that to solve the more general CSSI problem (Definition 1), we need to consider the possibility of kernels generated by points of the form $P + [x]Q$ and $[y]P + Q$ where $x, y \in [1, 2^{e_1}]$

and $2 \mid y$. Thus, the search space has size $3 \cdot 2^{e_1-1}$. In our analysis of concrete costs, we restrict ourselves to generators of the form $P + [x]Q$ per the specifications of SIKE [2]. However, this design can be easily modified to apply to CSSI instances with identical an asymptotic cost. For reasons of space we do not go into details, but instead outline the procedure and cost analysis. Full details are available upon request and can be included in the full paper if required.

We follow [2, Sec. 1.1.3] to convert $\mathbb{F}_{p^2}$ arithmetic operations to $\mathbb{F}_p$ arithmetic — our basic unit of cost will be quantum circuits for $\mathbb{F}_p$ (modular) arithmetic. It is important to note that whilst quantum circuits for modular addition can be performed in-place (one of the registers is overwritten with the output), modular multiplication and inversion is performed out of place. This gives us the quantum primitives (where $a, b \in \mathbb{F}_p$) $U_{add} |a\rangle |b\rangle \mapsto |a\rangle |a+b\rangle$, $U_{\text{mult}} |a\rangle |b\rangle \left|0^{\lceil \log_2 p \rceil}\right\rangle \mapsto |a\rangle |a\rangle |a \cdot b\rangle$ and $U_{\text{invert}} |a\rangle \left|0^{\lceil \log_2 p \rceil}\right\rangle \mapsto |a\rangle \left|a^{-1}\right\rangle$ — all of these operations require a number of ancilla qubits to implement efficiently. These ancilla begin and end in a clean ($|0 \ldots 0\rangle$) state. Cost estimates for these quantum circuits in the Clifford+T quantum gate set were kindly provided [12] which we use in our experiments in Appendix C.2.

SIKE function	Total Ops			Depth			# of qubits	
	Add	Mult	Invert	Add	Mult	Invert	Total	Ancillas
`xDBL`	90	32	0	51	22	0	$32\log(p) + 18$	$12\log(p) + 18$
`4_iso_curve`	30	8	0	23	8	0	$23\log(p) + 9$	$6\log(p) + 9$
`4_iso_eval`	68	28	0	28	9	0	$70\log(p) + 54$	$36\log(p) + 54$
`xDBLADD`	168	64	0	60	28	0	$76\log(p) + 54$	$36\log(p) + 54$
`jInvariant`	111	44	1	64	24	1	$44\log(p) + 27$	$18\log(p) + 27$

Table 1: # of quantum $\mathbb{F}_p$ operations for the classically defined SIKE functions.

The cost of step 2) dominates the entire computation, both in quantum memory usage and gate-cost, hence we first compute the total number of qubits required by 2) and 3). We then designed a quantum circuit corresponding to Algorithm 8 of [2] which takes the upper-bound on the amount of quantum memory available and uses it in a greedy strategy to conserve quantum circuit-size, only uncomputing quantum registers to save space when the storage limit was hit. After the initial point $|R = P + [x]Q\rangle$ was constructed, we uncomputed all ancilla qubits leaving only the $|x\rangle |R\rangle$ and then begun the algorithm for computing a degree 2^e isogeny from R using the strategy of [14] which corresponds to step 2).

For simplicity, we assume e_1 is even, but the method is easily adapted if not. We choose to create isogenies of degree-4 rather than degree-2 in the following discussion for reasons of efficiency. The method of [14, Sec. 4.2.2] computes the curve $E' = E_1/\langle R \rangle$ which is the image of E_1 by the isogeny ϕ of kernel $\langle R \rangle$ together with $\phi = \phi_{e_1/2-1} \circ \ldots \circ \phi_0$ (as a composition of degree-4 isogenies). Along the way, curves E'_i are create for $i = 0, \ldots, e_1/2 - 1$ where $E'_0 = E_1$ and $\phi_i : E'_i \to E'_{i+1}$. Intermediate isogenies ϕ_i are defined by the 4-torsion points

$[4^{e_1/2-1-i}]R_i \in E_i[4]$ where $R_{i+1} = \phi_i(R_i)$, $R_0 = R$. A low memory quantum implementation of this procedure costs a circuit-size $O(e_1^2)$. It consists in computing the R_i and $[4^{e_1/2-1-i}]R_i$ sequencially. Another strategy has a quantum circuit-size of $O(e \log_2 e)$ and corresponds to the optimal classical strategy of [14, Sec. 4.2.2]. It consists in computing all the $[4^{e_1/2-1-i}]R_i$ in a different order. The tree structure of Figure 1 illustrates this. The root is R_0, and each left move is a multiplication by 4 while a left move is the computation of ϕ_i and its evaluation on the current point. In Appendix A, we discuss the concrete issues encountered to turn this classical procedure into a quantum circuit.

(a) Low storage strategy (b) Low compuation strategy

Fig. 1: Different strategies to traverse the isogeny tree

4 Parallelism and quantum search

One well-known strategy to parallelize Grover's algorithm to search the space $\{0,1\}^n$ with 2^s parallel quantum computers is to use *inner parallelism* [16]. This consists in dividing the search-space into 2^s paritions by assigning each quantum computer to search a space of $n-s$ bits with the first s bits unique to each quantum computer. A well-known result by Zalka [24] proves that this is essentially optimal when quantum oracle is treated as a black-box, giving us that Grover's algorithm does not exhibit the same benefits from parallelism as classical search. Whilst this strategy reduces the *individual* total circuit-depth and circuit-size by a factor of $O(2^{-\frac{s}{2}})$, the *total* quantum circuit-size over all of the 2^s quantum computers increases by a factor of $O(2^{\frac{s}{2}})$.

The results of [15] provide concrete estimates for the cost of solving the CSSI problem using Grover's algorithm under this assumption concerning parallelism. Yet, Zalka's results are only proven relative to treating the quantum oracle as a black-box. In reality, there is a great deal of structure in many quantum oracles which can be used to reduce the total cost of the quantum search procedure and in this section, we demonstrate how the results of [3] can be transformed into a parallel version which affords the same benefits in overhead reduction and offers a quantum circuit-size over all quantum computers with a penalty of only $O(2^{\frac{s}{4}})$ at the cost of requiring $O(2^{\frac{3}{2}s})$ classical resources. The strategy is essentially a hybrid of [3, Th. 4.3], [6, Alg. 5], and [13, Prop. 1], which all involve classical

preprocessing. Note that the below strategy collapses to that of [3] when we use a single quantum computer (ie. $s = 0$), giving a complexity of $O\big(2^{\frac{e}{2}} \cdot \sqrt{C_{f_e} \log_2 p}\big)$ (where C_{f_e} is the cost of a quantum circuit that computes a degree 2^e isogeny-path starting at a specific curve) whereas the Grover-based approach from [15] has a complexity of $O\big(2^{\frac{e}{2}} C_{f_e}\big)$.

Theorem 2 below proposes a different parallelism strategy to that offered by simply using Grover's algorithm. The entire process builds upon the idea of [3] of exploiting preprocessing to create a list of j-invariants corresponding to isogenies of degree 2^{e_2} starting from E_2, but partitions these j-invariants amongst 2^s quantum computers by using a strategy similiar to that in [6]. This partitioning of the sublists (or buckets) is performed according to their first s-bits and each quantum computer is assigned a unique bucket. The quantum search process leverages two quantum oracles to search for a degree-2^{e_1} isogeny — one for a cheap test that any identifies any $x \in \{0,1\}^{e_1}$ corresponding to an isogeny whose j-invariant matches the first correct s-bits and one an expensive test that checks whether these $x \in \{0,1\}^{e_2}$ correspond to exactly the isogeny we are search for. This draws upon the work of [17] and [6] which explore such strategies and how to balance the cost of calling different quantum oracles.

Theorem 2 (Solving the CSSI problem via quantum search). *Let the CSSI problem be defined by the promise that there exists a degree-2^e isogeny between two given elliptic curves E_1 and E_2 over $\mathbb{F}_{p^2}$. Then there exists a quantum algorithm that solves the CSSI problem defined by these parameters with probability close to 1 which exploits 2^s quantum computers, allowing s up to $O(2^{\frac{2}{3}e})$, requiring an asymptotic quantum circuit-size per quantum computer of*

$$O\big(2^{-\frac{3}{4}s} 2^{\frac{e}{2}} \sqrt{C_{f_e} \log_2 p}\big) \tag{4}$$

and both classical computation and storage respectively on the orders of

$$O\big(2^{\frac{3}{2}s} \sqrt{\frac{C_{f_e}}{\log_2 p}} M(p)\big) \quad \text{and} \quad O\big(2^{\frac{3}{2}s} \sqrt{C_{f_e} \log_2 p}\big). \tag{5}$$

where C_{f_e} is the number of quantum gates required to implement quantum circuits that evaluates a degree 2^e isogeny-path starting at E_1 and $M(p)$ is the cost of classical modular multiplication over $\mathbb{F}_p$.

Proof: We first sketch the algorithm and choose the optimal parameters after the algorithm is explained. For notation we assign each of the 2^s quantum computers a unique index $S \in \{0,1\}^s$ (ie. $S_0, S_1, \ldots, S_{2^s-1}$). Explicit formulae for the computational lower-bounds can be easily obtained (and are available in the attached scripts) by use of Lemma 1 combined with the Chernoff bound.

1) We first compute and store the j-invariants of all degree-2^{e_2} isogenies starting from the curve E_2, sorting these as they are generated into 2^s buckets $L_0, \ldots, L_{2^s-1}$ such that the index of the bucket matches the first s bits of

all its members. Under the mild assumption that j-invariants and uniformly randomly distributed, each bucket will therefore contain $\approx 2^{e_2-s}$ elements of size $2\lceil \log_2 p \rceil$ and by a simple application of the Chernoff-bound and the union bound we have a bound on the size of these buckets. Formally, we have that for all $i \in \{0, 1, \ldots, 2^s - 1\}$ it holds that $|L_i| \in \Big((1-\delta_2)2^{e_2-s}, (1+\delta_2)2^{e_2-s}\Big)$ for $0 < \delta < 1$ with probability at most $2^s 2 \exp\Big(-\frac{2^{e_2-s}\delta_2^2}{3}\Big)$. In the anomalous case that buckets are outside of this range, the algorithm can be easily adapted by combining smaller buckets by the mapping $L_i, L_j \rightarrow L_{i,j}$ or by splitting larger buckets by the mapping $L_i \rightarrow L_{i\|0}, L_{i\|1}$. We make the assumption we have 2^s buckets all within the above bound and return to the choice of δ_2 later.

By the assumption there exists a unique claw, there will be a single $x_* \in \{0,1\}^{e_1}$ and a unique bucket L_{S_*} such such that $f_{e_1}(x_*) \in L_{S_*}$. Each quantum computer S will execute the quantum circuit for f_{e_1}, but exploits a different choice of bucket L_S, hence we need to worry about the total cost with respect to all quantum computers as we do not know which is the correct bucket, but we are only concerned about the lower-bound for the success probability assuming we have chosen the correct label S_* and hence the correct bucket L_{S_*}.

2) For each index $S \in \{0,1\}^s$ we define the $e_1 + w + 1$ qubit quantum algorithm $\mathcal{A}_{e_1}^S(k) = Q(H^{\otimes e_1}, \mathcal{O}_{\chi_{S,e_1}}, k)$, where w is the number of ancilla qubits for implement the quantum evaluation $\mathcal{E}_{f_{e_1}}$. The algorithm $\mathcal{A}_{e_1}^S(k)$ is a simple application of amplitude amplification (see Theorem 1) to boost the success probability of the Hadamard transform on e_1 qubits relative to $\chi_{S,e_1} : \{0,1\}^{e_1} \longrightarrow \{0,1\}$ where

$$\chi_{S,e_1}(x) = \begin{cases} 1 & \text{if the first } s \text{ bits of } f_{e_1}(x) \text{ equal the index } S \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Under the assumption we can implement $\mathcal{E}_{f_{e_1}}$ using $e_1 + w$ qubits, we can easily create the quantum bit oracle $\mathcal{O}_{\chi_{S,e_1}}$ using one additional qubit via the serial application of $\mathcal{E}_{f_{e_1}}$, a layer of at most s X gates, a single $\wedge_s(X)$ gate, another layer of at most s X gates and the application of $\mathcal{E}_{f_{e_1}}^\dagger$. The X and $\wedge_s(X)$ components require only $O(s)$ quantum gates, a cost dominated by that for $\mathcal{E}_{f_{e_1}}$. We therefore have that the total quantum cost of $\mathcal{A}_{e_1}^S(k)$ is at most

$$C_{\mathcal{A}_{e_1}^S(k)} = k \cdot \Big(2C_{\mathcal{E}_{f_{e_1}}} + C_{\wedge_s(X)} + 2C_{X^{\otimes s}} + C_{\mathcal{O}_{\bar{n}}}\Big) + (2k+1) \cdot C_{H^{\otimes n}}. \tag{7}$$

The success probability of $\mathcal{A}_{e_1}^{S_*}(k)$ relative to $\chi_{S_*,e_1} : \{0,1\}^{e_1} \longrightarrow \{0,1\}$ is dependent upon $|\chi_{S,e_1}^{-1}(1)|$. Applying the Chernoff-bound again gives us that $|\chi_{S_*,e_1}^{-1}(1)| \in \Big(1-\delta_1)2^{e_1-s}, (1+\delta_1)2^{e_1-s}\Big)$ with probability $\geq 1 - 2\exp\Big(-\frac{2^{e_1-s}\delta_1^2}{3}\Big)$. We could also adapt the algorithm to use a preprocessing step that involves estimating $\frac{|\chi_{S,e_1}^{-1}(1)|}{2^{e_1}}$ via quantum amplitude estimation [5] to the desired degree of accuracy for each quantum computer (a feasible strategy given the parameters involved) but we simply make the assumption that $|\chi_{S_*,e_1}^{-1}(1)|$ is bounded.

By choosing $k_a = \left\lfloor \frac{\pi}{4 \arcsin \sqrt{2^{-s}}} \right\rfloor$, we therefore have that the success probability of $\mathcal{A}_{e_1}^S(k_a)$ relative to $\chi_{S,e_1} : \{0,1\}^{e_1} \longrightarrow \{0,1\}$ is

$$a_{e_1}^S(k_a) = \sin^2\left(\left(2 \left\lfloor \frac{\pi}{4 \arcsin \sqrt{2^{-s}}} \right\rfloor + 1\right) \arcsin \sqrt{\frac{|\chi_{S,e_1}^{-1}(1)|}{2^{e_1}}} \right) \tag{8}$$

which is in the approximate range of $\sin^2\left(\frac{\pi}{2}\sqrt{1 \pm \delta_1}\right) \approx 1$ when $\delta_1 \ll 1$ and whose lower-bound can be derived computationally as we know k_a, a and δ_1.

3) For each quantum computer S, we let $\mathcal{B}_{e_1}^S(k_b) = Q(\mathcal{A}_{e_1}^S(k_a), \mathcal{O}_{\chi_{e_2}}^S, k_b)$ where $\mathcal{O}_{\chi_{e_2}}^S$ is defined by $\chi_{S,e_2} : \{0,1\}^{e_1} \longrightarrow \{0,1\}$

$$\chi_{S,e_2}(x) = \begin{cases} 1 & \text{if } f_{e_1}(x) \in L_S \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The quantum oracle $\mathcal{O}_{\chi_{e_2}}^S$ can easily be constructed by a similar process as for $\mathcal{O}_{S,e_1}$ where we apply the serial application of $\mathcal{E}_{f_{e_1}}$, at most $(|L_S|+1)2\lceil \log_2 p \rceil$ X gates and $|L_S|$ $\wedge_{2\lceil \log_2 p \rceil}(X)$ gates to compute the membership test and one application of $\mathcal{E}_{e_1}^\dagger$ to uncompute. Explicitly, the membership test will use X gates to ensure that the space where the j-invariant is written is encoded to $1^{2\lceil \log_2 p \rceil}$ if $f_{e_1}(x) \in L_S$ and the $\wedge_{2\lceil \log_2 p \rceil}(X)$ gates XOR 1 onto the output space of the quantum bit oracle if we have that $f_{e_1}(x) \in L_S$. The cost of $\mathcal{B}_{e_2}^S(k_b)$ is therefore

$$\begin{aligned} C_{\mathcal{B}_{e_2}^S(k_b)} = k_b \Big(& 2C_{\mathcal{E}_{f_{e_1}}} + |L_S| C_{\wedge_{2\lceil \log_2 p \rceil}(X)} + (|L_S|+1) 2 C_{X^{\otimes 2\lceil \log_p \rceil}} + C_{\mathcal{O}_{\bar{n}}} \Big) \\ & + (2k+1) C_{\mathcal{A}_{e_1}^S(k_a)}. \end{aligned} \tag{10}$$

By the fact $\mathcal{A}_{e_1}^{S_*}(k_a)$ results in an $x \in \{0,1\}^{e_1}$ such that $\chi_{S_*,e_1}(x) = 1$ with probability $a_{e_1}^{S_*}(k_a)$ and by the unique claw assumption, we have $\mathcal{A}_{e_1}^{S_*}(k_a)$ succeeds with probability $\frac{a_{e_1}^{S_*}(k_a)}{|\chi_{S_*,e_1}^{-1}(1)|} \approx 2^{e_1 - s}$ with respect to $\chi_{S_*,e_2} : \{0,1\}^{e_1} \longrightarrow \{0,1\}$.

We therefore make a choice of $k_b = \left\lfloor \frac{\pi}{4 \arcsin \sqrt{2^{-(e_1 - s)}}} \right\rfloor$ which gives us that the success probability of $\mathcal{B}_{e_1}^{S_*}(k_b)$ relative to $\chi_{S_*,e_2} : \{0,1\}^{e_1} \longrightarrow \{0,1\}$ is

$$b_{e_2}^{S_*}(k_b) = \sin^2\left(\left(2 \left\lfloor \frac{\pi}{4 \arcsin \sqrt{2^{-(e_1 - s)}}} \right\rfloor + 1\right) \arcsin \sqrt{\frac{a_{e_1}^{S_*}(k_a)}{|\chi_{S_*,e_1}^{-1}(1)|}} \right) \tag{11}$$

which is approximately in the range of $\sin^2\left(\frac{\pi}{2}\sqrt{\frac{1 \pm \delta_1}{1 \pm \delta_1}}\right)$, hence when $\delta \ll 1$ we have a good probability of success. Again, given the parameters e_1, s and δ_1, an explicit lower-bound can be computationally provided, which must be multiplied through by the Chernoff-bound factor $1 - 2\exp\left(-\frac{2^{e_1 - s}\delta_1^2}{3}\right)$ as discussed in 2).

Quantum cost of the search procedure The full cost can be computed directly via the cost equations, but ignoring inexpensive gate contributions we have that the total cost per quantum computer will be

$$C_{\mathcal{B}^{S_*}_{e_2}(k_b)} \approx 2^s \frac{\pi}{4} 2^{\frac{e_1-s}{2}} \left(2^{e_2-s} C_{\wedge_{2\lceil \log_2 p \rceil}(X)} + 2\frac{\pi}{4} 2^{\frac{s}{2}} 2 C_{f_{e_1}} \right) \tag{12}$$

whilst the individual depth per quantum computer is at most approximately

$$\frac{\pi}{4} 2^{\frac{e_1-s}{2}} \left((1+\delta_2) 2^{e_2-s} C_{\wedge_{2\lceil \log_2 p \rceil}(X)} + 2\frac{\pi}{4} 2^{\frac{s}{2}} 2 C_{f_{e_1}} \right). \tag{13}$$

Optimisation of Equation (12) gives us that $e_2 \approx \frac{3}{2}s + \log_2\left(\frac{\pi C_{f_{e_1}}}{C_{\wedge_{2\lceil \log_2 p \rceil}(X)}}\right)$, hence substitution of this choice of e_2 and rewriting $e_1 = e - e_2$ in Equation (12) gives us an approximate cost per individual quantum computer of

$$C_{\mathcal{B}^{S_*}_{e_2}(k_b)} \approx 2^{-\frac{3}{4}s} 2^{\frac{e}{2}} \sqrt{4\pi^3 C_{f_{e_1}} C_{\wedge_{2\lceil \log_2 p \rceil}(X)}}, \tag{14}$$

so that we have an asymptotic depth of $O(2^{-\frac{3}{4}s} 2^{\frac{e}{2}} \sqrt{C_{f_{e-\frac{3}{2}s}} \log_2 p})$ per quantum computer and a total quantum circuit-size of $O(2^{\frac{s}{4}} 2^{\frac{e}{2}} \sqrt{C_{f_{e-\frac{3}{2}s}} \log_2 p})$. This gives us the same reduction of the overhead costs as in [3], but exhibits advantageous scaling with respect to parallelism compared to a Grover-based approach assuming the quantum oracle is a black-box, where we have that individual circuit-size and circuit-depth scales with $O(2^{-\frac{s}{2}})$ but total circuit-size scales with $O(2^{\frac{s}{2}})$.

Classical cost of the search procedure We require the computation of 2^{e_2} j-invariants, which can be performed relatively efficiently using the backtracking methods described in Section 3.2 of [1] and costs $\approx 65 \cdot 3 \cdot 2^{e_2-1}$ $\mathbb{F}_{p^2}$ multiplications. We also require at most $2^{e_2} \cdot 2\lceil \log_2 p \rceil$ classical bits of storage to store these j-invariants in their buckets. We therefore have that whilst the quantum part of the computation scales favourably compared to Grover's algorithm, this comes at the cost of classical precomputation and storage which scale with $O(2^{\frac{3}{2}s})$. □

5 Conclusions

We have studied the cost of solving the CSSI problem using the claw-finding paradigm in conjunction with Grover's algorithm under both realistic assumptions concerning the cost of the quantum oracle and both constraints upon the maximum allowable quantum circuit-depth and classical/quantum circuit-width. We have directly impacted upon the tables in the SIKE specification [2] generated using the methods from [15] which take into account the aforementioned constraints.

We have not impacted upon the optimal parameters for using Grover's algorithm to attack the CSSI problem via claw-finding [3] in the case where we have no constraints upon our resources. However, this is neither a realistic scenario or an acceptable benchmark for current trends in cryptanalysis. The NIST post-quantum standardisation process [20] imposes a restriction upon the maximum quantum circuit-depth [21] whilst both studies on the cryptanalysis of SIKE [1, 9, 15] and the SIKE design specification [2] pay particular attention to the scenario where we are working with a constraint upon the classical circuit-width and/or quantum circuit-width. Indeed, if there were no such constraints then a simple classical meet-in-the-middle approach would easily beat Grover's algorithm as we discussed in Section 2.

SIKE p434							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover [15, 2]	175	79	96	-	-	-	-
Tani [15, 2]	160	78	96	-	-	-	-
VW [15, 2]	142	56	96	-	-	-	-
Theorem 3	141	80	60	106	96	131/86	50/0
Theorem 3	160	72	96	106	96	131/86	50/36

SIKE p751							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover [15, 2]	256	160	96	-	-	-	-
Tani [15, 2]	240	159	96	-	-	-	-
VW [15, 2]	263	178	96	-	-	-	-
Theorem 3	222	173	58	105	96	291/85	48/0
Theorem 3	240	155	96	105	96	291/85	48/37

Table 2: Classical/quantum MAXWIDTH = 2^{96} with conservative costs.

SIKE p434							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover	191	103	96	-	-	-	-/78
Grover [3]	177	98	95	55	39	109/29	-/79
Theorem 3	147	110	56	111	96	131/86	37/-
Theorem 3	167	90	96	111	96	91/86	37/40

SIKE p751							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover	272	186	96	-	-	-	-/76
Grover [3]	257	180	95	57	42	269/31	-/76
Theorem 3	228	193	55	111	96	291/85	-/35
Theorem 3	248	173	95	111	96	291/85	40/35

Table 3: Classical/quantum MAXWIDTH = 2^{96} with realistic costs.

The issue of how we can exploit problem-specific structure to bypass assumptions concerning black-box properties is an interesting area, as are the assumptions that we make with regards to choosing cryptographic parameters. Whilst our approach scales badly with regard to classical circuit-size and storage, this form of parallelism appears to be extremely effective for small s and can be applied in conjunction with the naive approach of partitioning the search-space by simply fixing bits. Open problems include an extensive analysis of the trade-offs we can make with this method, an examination of how we might reduce the classical costs or freely trade storage for circuit-size by regenerating j-invariants as required, investigating new ways for the expensive quantum oracle to leverage precomputation and the extension of these methods to similar problems.

References

[1] Adj, G., Cervantes-Vázquez, D., Chi-Domínguez, J.J., Menezes, A., Rodríguez-Henríquez, F.: On the cost of computing isogenies between supersingular elliptic curves. In: International Conference on Selected Areas in Cryptography. pp. 322–343. Springer (2018)

[2] Azarderakhsh, R., Campagna, M., Costello, C., Feo, L., Hess, B., Jalali, A., Jao, D., Koziel, B., LaMacchia, B., Longa, P., et al.: Supersingular isogeny key encapsulation. Submission to the NIST Post-Quantum Standardization project (2017)

[3] Biasse, J.F., Pring, B.: A framework for reducing the overhead of the quantum oracle for use with grover's algorithm with applications to cryptanalysis of SIKE (2019 (accepted)), http://www.lix.polytechnique.fr/Labo/Jean-Francois.Biasse/papers/mathcrypt-grover-2019.pdf

[4] Boyer, M., Brassard, G., Høyer, P., Tapp, A.: Tight bounds on quantum searching. arXiv quant-ph/9605034 (1996)

[5] Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics 305, 53–74 (2002)

[6] Chailloux, A., Naya-Plasencia, M., Schrottenloher, A.: An efficient quantum collision search algorithm and implications on symmetric cryptography. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 211–240. Springer (2017)

[7] Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. J. Cryptology 22(1), 93–113 (2009), https://doi.org/10.1007/s00145-007-9002-x

[8] Costache, A., Feigon, B., Lauter, K.E., Massierer, M., Puskás, A.: Ramanujan graphs in cryptography. CoRR abs/1806.05709 (2018), http://arxiv.org/abs/1806.05709

[9] Costello, C., Longa, P., Naehrig, M., Renes, J., Virdia, F.: Improved classical cryptanalysis of the computational supersingular isogeny problem. IACR Cryptology ePrint Archive 2019, 298 (2019)

[10] Gidney, C.: Halving the cost of quantum addition. Quantum Journal (2018)

[11] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proc. of the 28$^{\text{th}}$ annual ACM symp. on Theory of computing. pp. 212–219. ACM (1996)

[12] Häner, T., Jaques, S., Naehrig, M., Roetteler, M., Soeken, M.: Submission #59 to pqcrypto 2020. private communication and available to PQCRYPTO 2020 committee members

[13] Hosoyamada, A., Sasaki, Y.: Quantum demiric-selçuk meet-in-the-middle attacks: applications to 6-round generic feistel constructions. In: International Conference on Security and Cryptography for Networks. pp. 386–403. Springer (2018)

[14] Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: International Workshop on Post-Quantum Cryptography. pp. 19–34. Springer (2011)

[15] Jaques, S., Schanck, J.M.: Quantum cryptanalysis in the ram model: Claw-finding attacks on sike. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019. pp. 32–61. Springer International Publishing, Cham (2019)

[16] Kim, P., Han, D., Jeong, K.C.: Time–space complexity of quantum search algorithms in symmetric cryptanalysis: applying to aes and sha-2. Quantum Information Processing 17(12), 339 (2018)

[17] Kimmel, S., Yen-Yu Lin, C., Han-Hsuan, L.: Oracles with costs. 10th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2015, May 20-22, 2015, Brussels, Belgium 44 (2015)

[18] Maslov, D.: Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization. Physical Review A 93(2), 022311 (2016)
[19] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2010)
[20] of Standards, N.I., Technology.: Nist project for post-quantum cryptography standardization. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography` (2016), accessed: 07/10/2018
[21] of Standards, N.I., Technology.: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. (2016)
[22] Tani, S.: An improved claw finding algorithm using quantum walk. In: International Symposium on Mathematical Foundations of Computer Science. pp. 536–547. Springer (2007)
[23] Van Oorschot, P.C., Wiener, M.J.: Parallel collision search with application to hash functions and discrete logarithms. In: Proceedings of the 2nd ACM Conference on Computer and Communications Security. pp. 210–218. ACM (1994)
[24] Zalka, C.: Grover's quantum searching algorithm is optimal. Physical Review A 60(4), 2746 (1999)

A Reversible traversal of the isogeny tree

As we are working with out-of-place multiplication and we cannot delete information as we could classically (though a measurement-based uncomputation strategy [10] could improve our work). The low storage strategy given in figure 1 (a) requiring a maximum storage cost of $e_1/2 + 1$ $\mathbb{F}_{p^2}$ points and one isogeny curve. This is achieved by traversing down the left-hand side of the tree by repeated doublings, stopping halfway down, uncomputing all but the last computed point, then continuing left down to the bottom of the tree from the one remaining point. Once a leaf is reached, the isogeny curve is computed, which allows us to move the point at the top of the tree one point right. The curve is then removed and the points created by the traversal of the left-hand side are removed by the same process used to compute them. This process is repeated recursively all the way down the right-hand side of the tree storing only the points $R_0, R_1, ..., R_n$ and the final curve computation. By only computing halway down and reversing, we never store more than $e_1/2 + 1$ points.

The second strategy, figure 1 (b), was constructed to reduce the computational cost of the circuit. In this strategy, we determined the doubling cleanup should be done at each branch (i.e. classical store point), storing only this branch point before continuing down until the base point is computed. The base point is then used to construct a degree 4 isogeny curve which then evaluates a single point so the process can continue. Even though doublings are cheaper in Table 1, two doublings are required to reach the next lower level and thus also two points must be stored if no additional cleanup is done. Once a basepoint is reached, a curve is constructed and the closest point needing to be evaluated is evaluated. The curve is unconstructed, but the basepoint is left, to evaluate future points as necessary.

While doublings are reversed as soon as a storage point is reached in this strategy, isogeny evaluations are not reversed until the entire path has been computed down to a basepoint. Once a base point is reached by an isogeny evaluation, the sub-strategy to the left is reversed and cleaned up, only leaving behind the base points (for future isogeny curve constructions). This strategy continues and storage is cleared until traversing down the right-hand side of the graph begins. As we have stored all base points up to this point, storing all right-hand side points also would increase storage costs greatly. As the right-hand side is traversed using isogeny evaluations, these points are computed until a storage point is reached and then a round of cleanup is performed. This substrategy is computed as before until the entire tree is computed. This strategy ends while storing all n base points and any storage points down the right hand size, plus the final curve construction.

Using this construction, we determined the lowest computational cost comes when the strategy tree resembles the classical strategy tree with a cost ration of 2.5 to 1. Once this is completed, a j-invariant is computed on the final curve.

B Error in quantum search

The following is simply a generalisation of the methods from [4] and simplifies the analysis, allowing us to easily derive computational lower-bounds with our scripts to confirm that the success probability is approximately 1.

Lemma 1 (Error-analysis and amplitude amplification). *Let $\mathcal{A}$ be an arbitrary quantum algorithm that uses no measurements whose success probability is $a \in \big(a_-, a_+\big)$ relative to the function $\chi : \{0,1\}^n \longrightarrow \{0,1\}$.*
Then if $\frac{\arcsin\sqrt{a_+}}{\arcsin\sqrt{a_-}} + 2\arcsin\sqrt{a_+} \leq 2$ then the quantum amplitude amplification procedure $Q(\mathcal{A}, \mathcal{O}_\chi, k)$ where $k = \left\lfloor \frac{\pi}{4\arcsin\sqrt{a}} - \frac{1}{2} \right\rfloor$ succeeds with probability at least $\cos^2\Big(\arcsin\sqrt{a_+} + (2k+1)\big(\arcsin\sqrt{a_+} - \arcsin\sqrt{a_-}\big)\Big)$

Proof: In the following, $\theta_+ = \arcsin\sqrt{a_+}$, $\theta_- = \arcsin\sqrt{a_-}$ and $\theta_a = \arcsin\sqrt{a}$. Let $\hat{k} = \frac{\pi}{4\theta_a} - \frac{1}{2}$ and $k = \lfloor \hat{k} \rceil = \left\lfloor \frac{\pi}{4\theta_a} \right\rfloor$. By the choice of k we have that

$$\Big|\big(2\hat{k}+1\big)\theta_a - \big(2k+1\big)\theta_a\Big| \leq \theta_+ \tag{15}$$

and furthermore we know that for $\theta_- \leq \theta \leq \theta_+$

$$\Big|\big(2k+1\big)\theta_a - \big(2k+1\big)\theta\Big| \leq \big(2k+1\big)\big(\theta_+ - \theta_-\big). \tag{16}$$

Noting that $\big(2\hat{k}+1\big)\theta_a = \frac{\pi}{2}$ and applying the triangle inequality then gives us

$$\Big|\frac{\pi}{2} - \big(2k+1\big)\theta\big)\Big| \leq \theta_+ + \big(2k+1\big)\big(\theta_+ - \theta_-\big) \tag{17}$$

which by the condition $\frac{\theta_+}{\theta_-} + 2\theta_+ \leq 2$ ensures that as the LHS of Equation (17) is upper-bounded by $\frac{\pi}{2}$, hence taking sine of both sides, using the fact that $\sin(-x) = -\sin(x)$ and squaring gives us that

$$\sin^2\Big(\frac{\pi}{2} - \big(2k+1\big)\theta\big)\Big) \leq \sin^2\Big(\theta_+ + \big(2k+1\big)\big(\theta_+ - \theta_-\big)\Big). \tag{18}$$

Finally, multiplying through by -1 and the identities $\sin(\frac{\pi}{2} - x) \equiv \cos(x)$ and $\sin(x) \equiv 1 - \cos^2(x)$ give us that

$$\sin^2\Big(\big(2k+1\big)\theta\big)\Big) \geq \cos^2\Big(\theta_+ + \big(2k+1\big)\big(\theta_+ - \theta_-\big)\Big). \tag{19}$$

The result follows as $\sin^2\Big(\big(2k+1\big)\theta\big)\Big)$ is the probability of success of for the amplitude amplification procedure $Q(\mathcal{A}, \mathcal{O}_\chi, k)$ where $k = \lfloor \frac{\pi}{4\theta_a} \rfloor$. We note that the above upper-bound is not optimal, but provides an easy to check initial condition and allows us to consider general success probabilities of algorithms, rather than as a parameter of $|\chi^{-1}(1)|$ and $\mathcal{A}\,|0^n\rangle$. □

C Concrete estimates

C.1 Estimates using a conservative quantum oracle cost

In this section we examine the effect of our parallelism strategy using the conservative estimates for isogeny-circuits and $\mathbb{F}_{p^2}$ multiplications from [15]. These give the cost of the quantum circuit that computes a degree-2^e isogeny a cost of $e \log_2 e$ isogeny operations and the cost of these curve operations a conservative estimate of $4 \log_2 p \log_2 \log_2 p$ operations. Each quantum computer is assumed to use only $e_1 + 2 \log_2 p$ qubits and to assign classical costs, we also give each classical curve operation a cost of $4 \log_2 p \log_2 \log_2 p$ as we could always perform these individual curve operations on a small scale quantum computer if the algorithm exploits quantum properties. We exclude the cost of the parallel hardware to generate the table and the classical depth as this is a precomputation and could theoretically be aided by repurposing the hardware used to support the quantum error-correction as suggested in [15]. Scripts to produce these tables are provided in the supplementary material. The authors would like the reviewers to consider these conservative costs — we believe they are fair extension of [15].

As the MAXWIDTH constraint demonstrates how far we can reduce the depth of the quantum computation and we hit the bound for classical storage before we hit the bound for quantum storage, we allow for each quantum computer to employ a standard Grover parallelism strategy with black-box scaling of quantum circuit-depth being reduced by $O\big(2^{-t/2}\big)$ whilst total quantum circuit-size grows with $O\big(2^{t/2}\big)$ if the search-space of each original quantum computer is further split amongst 2^t quantum computers. This does not affect the classical costs.

SIKE p434							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover [15, 2]	158	96	63	-	-	-	-
Tani [15, 2]	143	95	62	-	-	-	-
VW [15, 2]	155	95	70	-	-	-	-
Theorem 3	140	96	51	96	86	141/76	43/0

SIKE p751							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover [15, 2]	320	96	224	-	-	-	-
Tani [15, 2]	304	95	224	-	-	-	-
VW [15, 2]	236	96	151	-	-	-	-
Theorem 3	247	95	158	247	238	149/227	147/0

Table 4: Constrained classical/quantum MAXDEPTH = 2^{96}.

SIKE p434							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover [15, 2]	175	79	96	-	-	-	-
Tani [15, 2]	160	78	96	-	-	-	-
VW [15, 2]	142	56	96	-	-	-	-
Theorem 3	141	80	60	106	96	131/86	50/0
Theorem 3	160	72	96	106	96	131/86	50/36

SIKE p751							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover [15, 2]	256	160	96	-	-	-	-
Tani [15, 2]	240	159	96	-	-	-	-
VW [15, 2]	263	178	96	-	-	-	-
Theorem 3	222	173	58	105	96	291/85	48/0
Theorem 3	240	155	96	105	96	291/85	48/37

Table 5: Constrained classical/quantum MAXWIDTH = 2^{96}.

MAXDEPTH With respect to a maximum quantum circuit-depth, we have that the performance of Algorithm 3 requires interpretation — the cost of the classical storage dominates, hence it can be argued that our algorithm does not offer a superior Depth×Width metric. This is an issue that requires further study — whilst each call to the expensive oracle requires access to the buckets of j-invariants, classical storage access patterns are clearly very different to accessing data through a quantum circuit, where we must pay the cost for operating in superposition by applying a deterministic circuit on all bits that might possibly be changed. If the elements of the individual buckets are sorted, subdivided into smaller storage devices and potentially stored in an efficient data structure such as a trie, then at most one of these substorage devices will be active at anytime, thereby reducing the total access-pattern cost. Nevertheless, classical storage costs are expected to far cheaper than quantum hardware and there is a balancing of costs that can be achieved here with respect to any real-world implementation. We leave an examination of this issue for future work.

MAXWIDTH As can be seen from the above tables, at least with respect to the conservative assumptions as stated above, Theorem 2 gives superior performance in the MAXWIDTH constrained scenario (a key assumption with respect to the security of SIKE p751 [1, 2]) both in the Gate Metric and Depth×Width metric for SIKE p434 and SIKE p751, offering both the best Gate cost, the best Depth×Width cost and an option giving it the same Gate and Width cost as Tani's algorithm whilst offering a superior Depth.

C.2 Estimates using a realistic cost analysis for the quantum oracle

In this section we assign the quantum oracle a realistic cost, as analysed in Section 3 to generate the quantum circuit-size, quantum circuit-depth in terms of $\mathbb{F}_p$ multiplications, additions and inversions as well as the quantum circuit-width. It is first worth noting that the conservative estimates with regards to our estimations and query-optimal parameters.

SIKE p434								
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover [15] (conservative costs)	132	122	10	-	-	109/108	-
Grover [3] (conservative costs)	126	116	10	37	22	205/15	-
Grover (realistic costs)	152	142	18	-	-	109/108	-
Grover [3] (realistic costs)	138	137	18	55	40	187/30	-

Table 6: Optimal parameters with no constraints for the naive Grover attack and it's extension [3] with our cost analysis of the quantum oracle.

We concern ourselves with only the classical circuit-size and storage requirements for the generation of the buckets of j-invariants, by the same discussion as in Ap-

pendix C.1 and use the estimation that we require $65 \cdot 32^{e_2-1}$ $\mathbb{F}_p$ multiplications. We again count the cost of building this table.

The hardware area for $\mathbb{F}_p$ multiplication is obtained from synthesizing a fairly optimized and parallelized hardware architecture for multiplier described in VHDL and implemented in ASIC based on 65-nm technology and is converted to gate equivalence (GE). Similar multiplier architecture has been used in hardware design of SIKE protocol [2]. Using this, we have that the GE for for p434 is 157,014 NAND gates and p751 is 277,704 NAND gates. The cost of the quantum circuits for modular arithmetic have been provided by [12].

SIKE p434							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover	197	96	110	-	-	63/62	-/92
Grover [3]	179	96	99	54	39	106/29	-/82
Theorem 3	152	96	73	137	121	105/112	-/55

SIKE p751							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover	359	96	271	-	-	-	-/253
Grover [3]	340	96	262	56	41	102/30	-/244
Theorem 3	-	-	-	-	-	-	-

Table 7: Constrained classical/quantum MAXDEPTH = 2^{96}.

SIKE p434							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover	191	103	96	-	-	-	-/78
Grover [3]	177	98	95	55	39	109/29	-/79
Theorem 3	147	110	56	111	96	131/86	37/-
Theorem 3	167	90	96	111	96	91/86	37/40

SIKE p751							
	Computation			Precomp		Params	
Attack	G	D	W	G	W	e_1/e_2	s/t
Grover	272	186	96	-	-	-	-/76
Grover [3]	257	180	95	57	42	269/31	-/76
Theorem 3	228	193	55	111	96	291/85	-/35
Theorem 3	248	173	95	111	96	291/85	40/35

Table 8: Constrained classical/quantum MAXWIDTH = 2^{96}.

As can be seen again, our technique offers a method to exploit classical computation and storage to provide superior parallelism. In the MAXDEPTH scenario this type of parallelism is limited, but still effective and can augment traditional approaches to realising parallelism with Grover's algorithm. We have have superior performance for our algorithm to all other current approaches for Grover's algorithm applied to cryptanalysis of SIKE when the MAXWIDTH constraint is enforced.

We note that in experiments the quantum circuit-size for the algorithm scales slightly better than the suggested $O(2^{s/4})$, as the cost of the circuit which computes the isogeny is reduced as s increases. This is a minor observative, but worth noting.

D Adaptations of Theorem 2 to constraints

In this section we note some features and adaptations that can be be made to the naive implementation of the algorithm as described in Theorem 2.

One precomputation — many attacks The SIKE specification [2] gives us that one of the curves will always be $E_1/\mathbb{F}_{p^2} : y^2 = x^3 + 6x + x$, hence choosing to precompute the buckets of j-invariants of all degree-2^{e_2} isogenies starting from this curve will work for any particular key-exchange instance. The precomputation may be started before network-traffic originating from a real world use of SIKE-p is captured, so long as p is known. We note this applies to purely classical MITM attacks based upon precomputing tables [1, 9] as well.

Sequential parallelism If we adhere to both the MAXDEPTH constraint and the MAXWIDTH constraint then we are limited in the parallelism we can employ to avoid the MAXDEPTH constraint in the naive implementation of the algorithm described in Theorem 2 and must be content with an algorithm with a success probably far lower than 1. If instead MAXDEPTH is interpreted as the maximum allowable quantum circuit-depth of any *individual* quantum computer used in cryptanalysis[4] then another strategy allows us to construct a hybrid quantum-classical search process with a success probability of close to one. This strategy is essentially is to execute only 2^k of the 2^s quantum computers at one time, which allows us to inherently bound the quantum circuit-width. Such a strategy clearly has no impact upon either the quantum Gate or Depth×Width metrics, but does have an impact upon both the total running time of the computation and the classical costs.

If the classical storage requirements are bounded, then a naive strategy is to simply recompute all 2^{e_2} j-invariants at each stage, keeping only the $\approx 2^{e_2-s+k}$ j-invariants that correspond to the 2^k quantum computers we are about to execute in the next layer of computation.

The naive sequential parallelism strategy therefore helps control how the classical storage costs scale as we require $O(2^{e_2-s+k+\log_2} \log_2 p) = O(2^{s/2+k} \log_2 p)$ storage at any one time, compared to $O(2^{3/2s}) \log_2 p$ if we had unbounded storage. As we execute 2^k quantum computers at a time, there are 2^{s-k} such interleaved layers of quantum computation combined with classical processing, leading to an increase of the classical circuit-size by a factor of 2^{s-k}. There is the possibility that this additional classical gate complexity can be reduced by an efficient data structure that keeps track of which bitstrings in the domain $\{0,1\}^{e_2}$ correspond to j-invariants which have been exploited by a previous bucket (thereby allowing us to avoid recomputing previously processed j-invariants), but we leave this for future investigations. Without using such a strategy, we must be content with an algorithm that may have succeed with probability $\ll 1$.

In the case that classical costs dominate, we can parallelise from the starting curve, in the case that quantum costs dominate we can parallelise from the end curve.

[4] an assumption rising from there being no restriction on classical circuit-depth