Research Article

Jean-François Biasse* and Benjamin Pring*

A framework for reducing the overhead of the quantum oracle for use with Grover's algorithm with applications to cryptanalysis of SIKE

https://doi.org/10.1515/jmc-2020-0080
Received Jun 05, 2019; accepted Jul 01, 2019

Abstract: In this paper we provide a framework for applying classical search and preprocessing to quantum oracles for use with Grover's quantum search algorithm in order to lower the quantum circuit-complexity of Grover's algorithm for single-target search problems. This has the effect (for certain problems) of reducing a portion of the polynomial overhead contributed by the implementation cost of quantum oracles and can be used to provide either strict improvements or advantageous trade-offs in circuit-complexity. Our results indicate that it is possible for quantum oracles for certain single-target preimage search problems to reduce the quantum circuit-size from $O\left(2^{n/2} \cdot mC\right)$ (where C originates from the cost of implementing the quantum oracle) to $O(2^{n/2} \cdot m\sqrt{C})$ without the use of quantum ram, whilst also slightly reducing the number of required qubits.
This framework captures a previous optimisation of Grover's algorithm using preprocessing [21] applied to cryptanalysis, providing new asymptotic analysis. We additionally provide insights and asymptotic improvements on recent cryptanalysis [16] of SIKE [14] via Grover's algorithm, demonstrating that the speedup applies to this attack and impacting upon quantum security estimates [16] incorporated into the SIKE specification [14].

Keywords: quantum search, reversible computation, quantum cryptanalysis

2010 Mathematics Subject Classification: 68Q12

1 Introduction

Whilst the quantum circuit-complexity of a quantum algorithm is linked to the cost of executing a quantum algorithm, this link is not yet fully understood owing to the uncertainty regarding the eventual architecture of quantum computers and the need to perform quantum error-correction to protect the state from environmental noise. The logical quantum circuit-model of computation ignores the issue of noise and has been the de-facto choice of assigning a cost to quantum algorithms for the cryptographic community as our understanding of the true costs involved with executing quantum algorithms has been evolving. In particular, there is the issue of *quantum query-complexity* versus *quantum bit-complexity* when assigning a cost to the best known quantum attack on a cryptosystem for purposes of choosing quantum-resistant cryptographic parameters in relation to it.

***Corresponding Author: Jean-François Biasse:** University of South Florida, United States of America; Email: biasse@usf.edu
***Corresponding Author: Benjamin Pring:** University of South Florida, United States of America;
Email: benjamin.pring@gmail.com

If a quantum search algorithm requires $O\left(2^{n/2}\right)$ calls to a particular subroutine (a quantum oracle), then it is clear that this algorithm has a cost of at least $O\left(2^{n/2}\right)$. If we assign a cost to this quantum oracle of C, then it is clear that the full cost of the algorithm is at least $O\left(2^{n/2} \cdot C\right)$. Whilst there are hard proofs that we cannot do better than $O\left(2^{n/2}\right)$ calls to this quantum oracle if we assume that the quantum oracle is a *black-box* [28] (in that we model it simply via input and output), we focus upon redefining what it means for the oracle to be called. By doing this, we note that for certain problems we can in fact *increase* the query-complexity but reduce the total cost of the quantum algorithm itself.

Contributions

We provide a framework for reasoning about how the quantum circuit-complexity of Grover's algorithm can be reduced via design principles that can be applied to the quantum oracle, allowing strict gains in all metrics for certain problems. This is done via combining classical search with Grover's algorithm, increasing the cost of the quantum oracle, but defining it over a smaller search-space. This approach allows for a balancing of the query-complexity and the cost of the quantum oracle and admits a number of benefits, such as preprocessing options which strictly improve the efficiency of Grover's algorithm.

We demonstrate the utility of our framework by applying it to two known quantum attacks on cryptosystems using Grover's algorithm, demonstrating that it captures and improves upon a known quantum attack on the Multivariate Quadratic problem over $\mathbb{F}_2$ using Grover's algorithm and provides new results on quantum cryptanalysis of SIKE [14], providing evidence that the cost of attacking SIKE via Grover's algorithm is asymptotically lower than previously estimated [14, 16].

Outline of this paper

In Section 2, we review Grover's algorithm. In Section 3 we introduce our framework. In Section 4 we examine several applications to cryptanalysis and give our conclusions in Section 5.

2 Background

Definition 2.1 (The unstructured search problem) Let $\chi : \{0,1\}^n \longrightarrow \{0,1\}$ be such that $M_\chi = |\chi^{-1}(1)|$. The *unstructured search problem* defined by χ is the problem of finding an element $x \in \{0,1\}^n$ such that $\chi(x) = 1$ or proving that no such element exists, given only the ability to evaluate χ.

A classical computer requires $O(\frac{2^n}{M_\chi})$ calls to a classical circuit which evaluates χ before a solution to the unstructured search problem (Definition 2.1) is found [1]. In comparison, Grover's algorithm requires $O(\sqrt{\frac{2^n}{M_\chi}})$ calls to a quantum circuit which evaluates χ and terminates with a solution to the unstructured search problem with high probability. It will additionally prove useful to consider another formulation of the search problem.

Definition 2.2 (The preimage search problem) Let $h : \{0,1\}^n \longrightarrow \{0,1\}^m$ and $Y_h \subseteq \{0,1\}^m$. The *preimage search problem* is to find an $x \in \{0,1\}^n$ such that $h(x) \in Y_h$ or prove that no such $x \in \{0,1\}^n$ exists.

Any algorithm that solves arbitrary instances of the preimage search problem can be used to solve the search problem and vice versa, but it is clear that there is more computational structure in the preimage search problem compared to the unstructured search problem which can benefit the design of of algorithms.

Quantum algorithms

Quantum states consist of *qubits* (*quantum bits*) and an n qubit quantum state relative to the *computational basis* $\{|x\rangle : x \in \{0,1\}^n\}$ can be expressed as $\sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$ where $\alpha_x \in \mathbb{C}$ and $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$. The α_x are the *amplitudes* of each computational basis state $|x\rangle$ and measurement of this quantum state results in the bitstring $x \in \{0,1\}^n$ with probability $|\alpha_x|^2$. Quantum algorithms therefore consist of increasing the magnitude of α_x which encode algorithmically useful information — Grover's algorithm consists of the repeated application of a quantum circuit, each of which (up to a point) increases the magnitude of α_x which encode solutions to the search problem.

Cost models and reversibility

Quantum circuits that do not include measurement are equivalent to unitary operators (U such that there exists $U^\dagger$ with the property $UU^\dagger = U^\dagger U = I$) and because of this correspondence, quantum circuits which implement $\chi : \{0,1\}^n \longrightarrow \{0,1\}$ can be designed by considering *reversible* classical circuits (which implement permutations and therefore all have inverses), with each reversible gate assigned a cost in terms of quantum gates.

Much as the universal boolean gate set $\{\neg, \oplus, \wedge\}$ can implement arbitrary classical circuits, quantum algorithms can be implemented (up to an arbitrary level of precision) by a *universal quantum gate set*. For reasons of space we deal only with asymptotics in this paper, but illustrate the above in terms of the Clifford+T universal quantum gate which consists of the Clifford gate set (the Hadamard, Phase and CNOT gates) and the single T gate. By fixing a universal quantum gate set we can reason about the quantum circuit-complexity (cost) of a quantum algorithm which consists of the quantum circuit-size (number of quantum gates), quantum circuit-depth (timesteps taken) and quantum circuit-width (quantum bits required). It is plain that the set of quantum gates $\{X, \wedge_1(X), \wedge_2(X)\}$ and more generally $\wedge_k(X)$ for $k \geq 1$ acting upon computational basis states defined by

$$X|x_1\rangle \mapsto |x_1 \oplus 1\rangle, \qquad \wedge_k(X)|x_1 \dots x_k\rangle |x_{k+1}\rangle \mapsto |x_1 \dots x_k\rangle \left|x_{k+1} \oplus (x_1 \wedge \dots \wedge x_k)\right\rangle \tag{1}$$

where $\wedge_0(X) := X$ is sufficient to implement all reversible classical circuits on computational basis states, if we have sufficient ancilla qubits as this gate set corresponds to the universal boolean gate set $\{\neg, \oplus, \wedge\}$. The $\wedge_k(X)$ for $k \geq 2$ is simply a useful abstraction. The X and $\wedge_1(X)$ gate each require one Clifford gate to implement, whilst the $\wedge_2(X)$ (Toffoli gate) can be implemented using 17 Clifford+T gates [2, 24] and the $\wedge_k(X)$ gate to require at most $40k - 64$ Clifford gates for $k > 2$ [17] if we have a single ancilla qubit, which can be in any state.

Definition 2.3 (Cost notation) If $\mathcal{A}$ is any quantum algorithm or quantum gate, we denote the execution cost of $\mathcal{A}$ by the notation $C_{\mathcal{A}}$. Costs will be provided in terms of components that are executed in serial, so that $C_{\mathcal{A}}$ can be substituted for circuit-size, circuit-depth or either metric applied to a subset of quantum gates.

2.1 Quantum oracles and Grover's algorithm

Definition 2.4 (Quantum bit oracle) The *quantum bit oracle* $\mathcal{O}_\chi^{(b)}$ acting upon $n+1$ qubit computational basis states $|x_1 \dots x_n\rangle |b\rangle$, where $b \in \{0,1\}$, maps

$$\mathcal{O}_\chi^{(b)} |x_1 \dots x_n\rangle |b\rangle \mapsto |x_1 \dots x_n\rangle \left|b \oplus \chi(x_1 \dots x_n)\right\rangle. \tag{2}$$

Quantum oracles will be used in conjunction with *Grover's algorithm*, which we state and provide a cost for without proof. Our modifications will simply be alterations of the quantum bit oracle and are used with Grover's algorithm.

Theorem 2.5 (Grover's algorithm [4, 12]) *Let $\chi : \{0,1\}^n \longrightarrow \{0,1\}$ define the search problem where $M = |\chi^{-1}(1)|$ is known. Then there exists a quantum algorithm that solves the search problem defined by χ with prob-*

ability at least $\max\{1-\frac{M}{2^n}, \frac{M}{2^n}\}$ *and which has a cost of* $C_{H^{\otimes n}} + \left\lfloor \frac{\pi}{4} \cdot \frac{2^{n/2}}{\sqrt{M}} \right\rfloor \cdot (C_\chi + C_{D_n})$, *where* $C_{H^{\otimes n}}$, C_χ *and* C_{D_n} *are respectively the cost of implementing the Hadamard transform on n qubits, the quantum bit oracle* $\mathcal{O}_\chi^{(b)}$ *and the diffusion operator* D_n *on n qubits, where cost is either quantum circuit-size or quantum circuit-depth.*

$H^{\otimes n}$ is the parallel application of n Hadamard gates, each of which cost 1 Clifford gate and the diffusion operator on n qubits is can be assigned a circuit-size of $44n - 105$ Clifford+T gates for $n \geq 7$ [17, 18] and circuit-depth of $44n - 103$. Our framework will enable the cost expressed in Theorem 2.5 to be optimised by trading off between the cost $C_\chi + C_{D_n}$ and the query-complexity term $\left\lfloor \frac{\pi}{4} \cdot \frac{2^{n/2}}{\sqrt{M}} \right\rfloor$. Much as we require memory to implement classical functions efficiently, we often require ancilla qubits to implement the *action* of quantum bit oracle. In this paper we use a decomposition of the quantum bit oracle that captures this fact.

Definition 2.6 (Bitwise decomposition of the oracle) A *bitwise decomposition* of quantum bit oracle $\mathcal{O}_\chi^{(b)}$ consists of the $n+1$ unitary operators $U_{\chi_\star}, U_{\chi_n}, \ldots, U_{\chi_1}$ acting upon $n+w+1$ qubits, such that for any $x_1 \ldots x_n \in \{0,1\}^n$ and $b \in \{0,1\}$

$$\begin{aligned} U_{\chi_1}^\dagger \cdots U_{\chi_n}^\dagger U_{\chi_\star} U_{\chi_n} \cdots U_{\chi_1} |g_0\rangle |x_1 \ldots x_n\rangle |b\rangle \\ \mapsto |g_0\rangle |x_1 \ldots x_n\rangle \left|b \oplus \chi(x_1 \ldots x_n)\right\rangle, \end{aligned} \tag{3}$$

where $U_{\chi_i} = U'_{\chi_i} \otimes \mathcal{I}^{\otimes n-i+1}$ so that U'_{χ_i} acts upon $w+i$ qubits, with

$$U'_{\chi_i} \left|g_{i-1}(x_1, \ldots, x_{i-1})\right\rangle |x_1 \ldots x_i\rangle \mapsto \left|g_i(x_1, \ldots, x_i)\right\rangle |x_1 \ldots x_i\rangle \tag{4}$$

with $g_i(x_1, \ldots, x_i) \in \{0,1\}^w$ derived from $x_1, \ldots, x_i$ only, $g_0 \in \{0,1\}^w$ and

$$U_{\chi_\star} \left|g_n(x_1, \ldots, x_n)\right\rangle |x_1 \ldots x_n\rangle |b\rangle \mapsto \left|g_n(x_1, \ldots, x_n)\right\rangle |x_1 \ldots x_n\rangle \left|b \oplus \chi(x_1 \ldots x_n)\right\rangle. \tag{5}$$

We there have that $\left(\mathcal{I}^{\otimes w} \otimes \mathcal{O}_\chi^{(b)}\right) = U_{\chi_1}^\dagger \cdots U_{\chi_n}^\dagger U_{\chi_\star} U_{\chi_n} \cdots U_{\chi_1}$ and that U_{χ_i} should be interpreted as producing a memory state $g_i(x_1, \ldots, x_i) \in \{0,1\}^w$ computed using only the first i bits of a possible solution to the search problem. The memory state $g_0 \in \{0,1\}^w$ can be considered as an initial memory-state which does not depend upon any of the bits $x_1, \ldots, x_n$. Typically, we can take $g_0 = 0^w$. This decomposition applies trivially to quantum bit oracles constructed using only reversible boolean primitives (we define $U_{\chi_i} = \mathcal{I}^{\otimes n+w+1}$ and $U_{\chi_\star} = \mathcal{O}_\chi^{(b)}$) but non-trivial decompositions may require special design. The single-target preimage search problem (see Definition 2.2) can be modelled by simply by setting $U_{\chi_n} \cdots U_{\chi_1}$ to compute $\left|h(x_1 \ldots x_n) \oplus 1^m\right\rangle$ and setting $U_{\chi_\star} := \wedge_m(X)$.

3 A framework for preprocessing

In this section we present our framework for optimising applications of Grover's algorithm via modifying quantum bit oracles to take advantage of classical search and preprocessing. Computational gains will be made possible via examining the role of memory in implementing the action of the quantum bit oracle and trading off between query-complexity and computational effort required to implement the action of the quantum bit oracle. With this in mind we can choose an integer $0 \leq k \leq n$ that defines a *cut* of the bitwise decomposition of the quantum bit oracle (see Definition 2.6), splitting it into three separate components so that

$$U_{n-k} := U_{\chi_{n-k}} \cdots U_{\chi_1}, \qquad U_k := U_{\chi_n} \cdots U_{\chi_{n-k+1}} \qquad \text{and} \qquad U_\star := U_{\chi_\star}. \tag{6}$$

3.1 Combining classical search with Grover's algorithm

Theorem 3.1 (Secondary classical search) *Given a cut of a quantum oracle parameterised by* $0 < k < n$*, we can implement a modified quantum bit oracle*

$$\mathcal{O}_{\chi'}^{(b)} |0^w\rangle |x_1 \dots x_{n-k}\rangle |0^k\rangle |c\rangle \mapsto |0^w\rangle |x_1 \dots x_{n-k}\rangle |0^k\rangle |c \bigoplus_{z_1 \dots z_k \in \{0,1\}^k} \chi(x_1 \dots x_{n-k} z_1 \dots z_k)\rangle \tag{7}$$

and whose cost is

$$C_{\mathcal{O}_{\chi'}^{(b)}} = 2 \cdot \sum_{i=1}^{n-k} C_{U_{\chi_i}} + 2 \cdot 2^k \left(\sum_{i=1}^{k} C_{U_{\chi_{n-k+i}}} \right) + 2^k \cdot C_{U_\star} + 2^k \cdot C_X. \tag{8}$$

Proof. We first execute U_{n-k} to compute

$$U_{n-k} \left|0^w\right\rangle |x_1 \dots x_{n-k}\rangle |0^k\rangle \mapsto \left|g(x_1, \dots, x_{n-k})\right\rangle |x_1 \dots x_n\rangle |0^k\rangle \tag{9}$$

then simply follow the procedure of executing the sequence $U_k^\dagger U_\star U_k$ on all possible assignments of the final k values of the search-space. This can be performed efficiently via using the k qubits following the register $|x_1 \dots x_{n-k}\rangle$ as additional input $z_1 \dots z_k \in \{0,1\}^k$ for $U_k^\dagger U_\star U_k$ and simply cycling through all possible values of $z_1 \dots z_k \in \{0,1\}^k$. If we use a *binary reflected Gray Code* [11], we can start in the state 0^k and cycle through all 2^k elements of $\{0,1\}^k$, ending in the state 10^{k-1} by flipping only a single bit at a time, which can be accomplished via using an X gate on the relevant qubit and if we wish to return the state to $|0^k\rangle$, then we need only execute an additional X gate for a total cost of 2^k X gates. After this, we simply execute the unitary $U_{n-k}^\dagger$, leaving us with the computational basis state

$$\left|0^w\right\rangle |x_1 \dots x_{n-k}\rangle |0^k\rangle |c \bigoplus_{z_1 \dots z_k \in \{0,1\}^k} \chi(x_1 \dots x_{n-k} z_1 \dots z_k)\rangle . \tag{10}$$

□

Corollary 3.2 *The modified quantum bit oracle* $\mathcal{O}_{\chi'}^{(b)}$ *as described in Theorem 3.1 can be used with Grover's algorithm defined on the search-space of* $n-k$ *qubits and terminates with an* $x \in \{0,1\}^{n-k}$ *that can be extended to a full solution with probability at least* $\left(1 - \frac{M(M-1)}{2^{n-k}}\right) \cdot \max\left\{1 - \frac{M}{2^{n-k}}, \frac{M}{2^{n-k}}\right\}$ *if* $1 \le M \le 2^{(n-k)/2}$.

Proof. This can easily be seen as the modified quantum oracle will mark any element $x_1 \dots x_{n-k} \in \{0,1\}^{n-k}$ such that $x_1 \dots x_{n-k} \in \{0,1\}^{n-k}$ can be extended to a full solution for some $z_1 \dots z_k \in \{0,1\}^k$. Hence $M' = |\chi'^{-1}(1)| = |\chi^{-1}(1)|$ if there are no collisions on the first $n-k$ bits of solutions, for which a standard lower bound exists. If $M = 1$, then there can obviously be no such collision. □

Such stategies are possible with classical computation, but require *state* to be stored. By their nature, reversible logic circuits store state *implicitly* and by using this fact we avoid increasing the number of qubits. There is no guarantee that a non-trivial advantageous cut will be possible, but we can simply follow a design heuristic where as much cost as possible is shifted towards U_{n-k}. As we can simply compute the costs $C_{U_{n-k}}$, C_{U_k} and $C_{U_\star}$ as a function of k, we can easily find an optimal k via numerical simulation of the costs involved (often a simple formula) on all values of $0 \le k \le n$, which is a negligible classical computation.

Example 3.3 We consider the case where $C_{U_{\chi_1}} = \dots = C_{U_{\chi_n}} = C_{U_{\chi_\star}}$ and these costs dominate that of the diffusion step, so that $C_{U_{n-k}} = (n-k)D$, $C_{U_k} = kD$ and $C_{U_\star} = D$ for some constant D. Choosing $k = \log_2 n$ and using Equation (8) in conjunction with the Theorem 2.5 gives us a cost of

$$\frac{\pi}{4} \cdot 2^{n/2} \cdot \frac{1}{\sqrt{n}} \cdot \left(2\left(n - \log_2 n\right) + n\left(2 \log_2 n + 1\right)\right) \cdot D \tag{11}$$

which gives us an asymptotic cost of $O\left(2^{n/2} \cdot n^{1/2} (\log_2 n) D\right)$ compared to using Grover with the unmodified oracle for an asymptotic cost of $O\left(2^{n/2} \cdot nD\right)$.

Corollary 3.4 (Evaluation via backtracking) *Let the conditions be as in Theorem 3.1. The same procedure can be implemented for a cost of*

$$C_{\mathcal{O}^{(b)}_{\chi'}} = 2 \cdot \sum_{i=1}^{n-k} C_{U_{\chi_i}} + 2 \cdot \sum_{i=1}^{k} \left(2^i \cdot C_{U_{n-k+i}} \right) + 2^k \cdot C_{U_{\chi_\star}} + 2^k \cdot C_X. \tag{12}$$

Proof. This can be easily seen as if we denote via X_i the application of an X gate to the i^{th} qubit of the search-space then each subsequence of unitary operators

$$U_\star U_{\chi_n} \cdots U_{\chi_{n-k+1}} X_{n-k+i} U^\dagger_{\chi_{n-k+1}} \cdots U^\dagger_{\chi_n} U_\star \tag{13}$$

that appears in the unitary U_k can be replaced by the subsequence

$$U_\star U_{\chi_n} \cdots U_{\chi_{n-k+i}} X_{n-k+i} U^\dagger_{\chi_{n-k+i}} \cdots U^\dagger_{\chi_n} U_\star. \tag{14}$$

□

Corollary 3.5 (Commuting bitwise invariant components) *Given a modified quantum bit oracle $\mathcal{O}^{(b)}_{\chi'}$ parameterised by $0 \leq k \leq n$ as in Theorem 3.1 such that $U_{\chi_{n-k+1}}, \ldots, U_{\chi_n}$ all commute and the action of each U_{χ_i} is invariant upon any choice of $z_j \neq z_i$, the cost of $\mathcal{O}^{(b)}_{\chi'}$ can be reduced to*

$$C_{\mathcal{O}^{(b)}_{\chi'}} = 2 \cdot \sum_{i=1}^{n} C_{U_{\chi_i}} + \sum_{i=1}^{k} \left(2^i \cdot C_{U_{\chi_{n-k+i}}} \right) + 2^k \cdot C_{U_{\chi_\star}} + 2^k C_X. \tag{15}$$

Proof. Again using the notation X_i for the application of an X gate to the i^{th}, we can adapt Theorem 3.1 by simply replacing any subsequence

$$U_\star U_{\chi_n} \cdots U_{\chi_i} \cdots U_{\chi_{n-k+1}} X_{n-k+i} U^\dagger_{\chi_{n-k+1}} \cdots U^\dagger_{\chi_i} \cdots U^\dagger_{\chi_n} U_\star \tag{16}$$

that appears in the unitary U_k by

$$U_\star U_{\chi_i} X_{n-k+i} U_{\chi_n} \cdots U_{\chi_{n-k+1}} U^\dagger_{\chi_{n-k+1}} \cdots U^\dagger_{\chi_n} U^\dagger_{\chi_i} U_\star \tag{17}$$

by the commuting property of each U_{χ_i} and invariance of the unitary sequence $U_{\chi_n} \cdots U_{\chi_{n-k+1}}$ upon the variable z_i. From there it is a simple matter to note that the inner unitaries cancel each other out and we must first fully compute the sequence $U_{\chi_n} \cdots U_{\chi_1}$ and end with the sequence $U^\dagger_{\chi_1} \cdots U^\dagger_{\chi_n}$. □

Example 3.6 We again consider the case where each unitary operator a cost of D as in Example 3.3, but where we can instead apply Theorem 3.5. The choice of $k = \log_2 n$ can now be seen to be optimal if we take the derivative of the full cost equation for Grover's algorithm with the modified quantum bit oracle. This gives an asymptotic cost for Grover's algorithm with this modified quantum bit oracle of $O\left(2^{n/2} \cdot n^{1/2} D\right)$, whereas Theorem 3.1 gave us a cost of $O\left(2^{n/2} \cdot n^{1/2} \log_2 nD\right)$ and the unmodified quantum bit oracle with Grover was $O\left(2^{n/2} \cdot nD\right)$.

3.2 Preprocessing the classical secondary-search procedure

We now turn to the benefits of preprocessing any of the previously described methods of secondary classical search.

Theorem 3.7 (Ancilla qubits allow shifting of unitary costs) *Any component of the circuit that computes $U_{\chi_{n-k+i}}$ for $1 \leq i \leq k$ that is dependent solely on $|x_1 \ldots x_{n-k}\rangle \left|z_1 \ldots z_j\right\rangle \left|g_{n-k+j}(x_1, \ldots, x_{n-k}, z_1, \ldots, z_j)\right\rangle$ for $0 \leq j < i$ can be computed and stored on ancilla qubits during the computation of $U_{\chi_{n-k+j}}$.*

Proof. The proof of this is trivial and relies solely upon the definition of the bitwise decomposition of the quantum bit oracle. □

In an ideal situation, the unitary costs will be shifted as much as possible to U_{n-k}.

Theorem 3.8 (Classical preprocessing allows strict gains) *Let $\mathcal{O}^{(b)}_{\chi'}$ be a modified quantum bit oracle parameterised by $0 < k < n$ as in Theorem 3.1. Then at the cost of classical storage space and/or classical preprocessing and without affecting the correctness of this algorithm, the quantum cost of $\mathcal{O}^{(b)}_{\chi'}$ can be reduced and is at worst unchanged, whilst we reduce the number of qubits required by k.*

Proof. We will create 2^i circuits for each $U_{\chi_{n-k+i}}$, each of which are hardcoded to assume that the bits $z_1 \dots z_i \in \{0,1\}^i$ are fixed. The first benefit is that as we are implicitly creating a circuit which is hardcoded with a choice of $z_1 \dots z_i$, we need not include these qubits or any qubits which interact *only* with them (and not $x_1 \dots x_{n-k}$ by any circuit-path) in the search-space or the w-bit memory-state.

The second benefit is in a reduction in the complexity of the individual circuits themselves. If we consider purely reversible circuits, then for any unitary U we have that if any z_i appears in the control qubits for $\wedge_k(U)$, then this can be hardcoded as a either a $\wedge_{k-1}(U)$ gate if $z_i = 1$ or removed completely if $z_i = 0$.

The third benefit is that further optimisations are possible in the sequence of hardcoded circuits $U_{\chi_{n-k+1}} \cdots U_{\chi_n}$ as a whole. If we consider a simple circuit constructed of multiple $\wedge_k(X)$ gates, all of which write to the same target qubit and where no cancellation is posible, then any hardcoding of these $\wedge_k(X)$ gates that results in a circuit with $r \wedge_{k'}(X)$ for $(k' < k)$ gates with identical controls allows them to be removed if r is even or replaced with a single gate if r is odd.

Thus if we allow for the preprocessing and additional storage or alternatively online computation then these hardcoded quantum circuits are no more expensive to execute and we can always reduce the number of qubits by k. □

We briefly mention that we could employ parallelism (communication costs allowing), whereby we compute U_{n-k}, then create 2^k copies of the resulting state and execute the sequence of unitaries $U_k^\dagger U_\star U_k$ upon each one. This strategy allows us to bypass some of the increase in circuit-size that is a hard-limit if we treat the quantum oracle as a black-box [28] as this increase only applies to C_{U_k} and $C_{U_\star}$.

4 Applications to Cryptanalysis

In this section we demonstrate that our framework captures one previously proposed attack using Grover's algorithm on Multivariate Quadratic cryptosystems, provides missing asymptotic analysis on its results and improves upon it. We conclude with demonstrating our methodology can be applied to recent quantum cryptanalysis [16] of the proposed quantum resistant cryptosystem SIKE [14].

4.1 The Multivariate Quadratic problem over $\mathbb{F}_2$

Definition 4.1 (The Multivariate Quadratic ($\mathcal{MQ}$) problem over $\mathbb{F}_2$) We define $f^{(1)}(x_1, \dots, x_n), \dots, f^{(m)}(x_1, \dots, x_n) \in \mathbb{F}_2[x_1, \dots, x_n]$ be m equations of degree two in n variables over the finite field of size 2. The *Multivariate Quadratic* ($\mathcal{MQ}$) problem over $\mathbb{F}_2$ is to find a solution vector $(x_1, \dots, x_n) \in (\mathbb{F}_2)^n$ such that

$$f^{(1)}(x_1, \dots, x_n) = \cdots = f^{(m)}(x_1, \dots, x_n) = 0. \tag{18}$$

Several quantum resistant signature schemes [13, 20] have been published which rely upon the hardness of solving the Multivariate Quadratic problem over $\mathbb{F}_2$. Whilst asymptotically more efficient algorithms exist [3, 9], a basic attack [23] using Grover's algorithm that was later optimised via preprocessing [21] is both captured and improved upon by our framework. We leave explicit details to Appendix A for reasons of space and to avoid duplication of preexisting work [21, 23].

This case-study provides important commentary upon the difficulty in choosing quantum resistant parameters in relation to Grover's algorithm as the initial quantum resistant parameters were suggested [20] in relation to the *query-complexity* of $O\left(2^{n/2}\right)$ for Grover's algorithm to solve the $\mathcal{MQ}$ problem over $\mathbb{F}_2$. After publication of an explicit design for a quantum bit oracle to use in conjunction with Grover's algorithm for this problem [23] which gave the quantum circuit-size $O\left(2^{n/2} \cdot mn^2\right)$ for Grover's algorithm, new parameters were suggested in a subsequent paper [19] in relation to this cost. These costs were also quoted in several specifications for quantum-resistant cryptosystems in the NIST competition [6, 8]. Our framework demonstrates that one optimisation [21] using preprocessing lowers the cost to $O\left(2^{n/2} \cdot mn^{3/2}\right)$ and that by using our framework this improves to $O\left(2^{n/2} \cdot mn\right)$ by using an additional $O(m \log_2 n)$ ancilla qubits. We discuss the problem of choosing quantum-resistant cryptographic parameters in relation to anything but the query-complexity of Grover's algorithm further in Section 5.

4.2 The Computational SuperSingular Isogeny (CSSI) problem

In this section we reexamine the cost of a Grover-based attack upon the quantum-resistant key encapsulation method SIKE [14], whereby Grover is used to attack the CSSI problem (see Definition 4.2) via searching for a unique collision between two functions. We demonstrate how this attack fits into, and can be improved upon by, our framework. We provide an asymptotically better attack using Grover's algorithm and new estimates for the hardness of solving the CSSI problem via Grover's algorithm under various constraints (see Appendix B). These results impact upon the estimates in [16] which are quoted in the SIKE specification [14].

This problem has previously studied in [16] where the authors argue that whilst Tani's algorithm [25] may be the most asymptotically efficient method to solve this problem in terms of query-complexity, once the implementation of the underlying quantum data structure and memory is taken into account, Grover's algorithm may be competitive with Tani's algorithm.

On the cost of computing an isogeny-path

Isogenies are morphisms that are rational maps between groups of points of elliptic curves. Their degree is that of their rational map structure, and they are uniquely determined by their kernel. Given the 2^e-torsion $E[2^e]$ of E, a degree-2^e isogeny uniquely corresponds to a $x_1 \dots x_e \in \{0,1\}^e$ via a choice of a (cyclic) kernel in $E[2^e]$. Given a kernel, the total cost of computing the corresponding 2^e-isogenous curve is in $O(e \log_2 e)$ elliptic curve operations [10].

Definition 4.2 (The Computational SuperSingular Isogeny problem[1] [15])
Let E_1, E_2 be two supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ such that there is a 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 an isogeny between E_1 and E_2.

Finding (up to isomorphism) a degree-2^e isogeny $\phi : E_1 \longrightarrow E_2$ can be solved by finding one degree-2^{e_1} isogeny $\phi_1 : E_1 \to E'$ and one degree-2^{e_2} isogeny $\phi_2 : E_2 \to E''$ such that $e = e_1 + e_2$ and E' is isomorphic to E''. The composition of isogenies $\phi = \overline{\phi_2} \circ \phi_1$ (where $\overline{\phi_2}$ is the dual-isogeny of ϕ_2) is then the degree-2^e isogeny we are searching for. Isomorphic classes of curves are identified by their j-invariant in $\mathbb{F}_{p^2}$. Hence we define $h_i : \{0,1\}^{e_i} \longrightarrow \mathbb{F}_{p^2}$ for $i = 1, 2$ so that $h_1(x_1 \dots x_{e_1})$ and $h_2(x_1 \dots x_{e_2})$ are the respective j-invariants of E' where $\phi_i : E_i \to E'$ corresponds to the kernel defined by $x_1 \dots x_{e_i} \in \{0,1\}^{e_i}$. Thus, if we find the collision $(x_1 \dots x_{e_1}, z_1 \dots z_{e_2}) \in \{0,1\}^{e_1} \times \{0,1\}^{e_2}$ such that $h_1(x_1 \dots x_{e_1}) = h_2(z_1 \dots z_{e_2})$, then we have solved the CSSI problem. As in [16] we work under the assumption that there is a single such isogeny $\phi : E_1 \longrightarrow E_2$ (hence there is one target in our search-space), which is justified under the arguments of [26].

Fitting the attack to our framework

When $e_1 \approx e_2 \approx e/2$ as suggested in [16], we obtain a constant time saving over the simple search case $e_1 = e, e_2 = 0$ as $2 \cdot \frac{e}{2} \cdot \log_2(e/2) = e(\log_2 e - 1)$. This does not impact the asymptotic complexity of the search procedure. In our framework, we define the initial unitary U_{n-k} (in this scenario $n = e$ and $k = e_2$) to compute (where $\hat{g}_{e_1}(x_1, \ldots, x_{e_1})$ is the intermediate memory-state required to compute the j-invariant $h_1(x_1 \ldots x_{e_1})$)

$$|\hat{g}_{e_1}(x_1, \ldots, x_{e_1})\rangle \; |0^{w_2}\rangle \; |h_1(x_1 \ldots x_{e_1})\rangle \, |x_1 \ldots x_{e_1}\rangle \, |z_1 \ldots z_{e_2}\rangle \tag{19}$$

where $\left|g_{e_1}(x_1, \ldots, x_{e_1})\right\rangle = \left|\hat{g}_{e_1}(x_1, \ldots, x_{e_1})\right\rangle \; |0^{w_2}\rangle \; |h_{e_1}(x_1 \ldots x_{e_1})\rangle$ in our framework and the unitary U_k (where $k = e_2$) is defined to map this state to

$$|\hat{g}_{e_1}(x_1, \ldots, x_{e_1})\rangle \left|\hat{g}_{e_2}(z_1, \ldots, z_{e_2})\right\rangle \; |h_1(x_1 \ldots x_{e_1}) \oplus \overline{h_2(z_1 \ldots z_{e_2})}\rangle \, |x_1 \ldots x_{e_1}\rangle \, |z_1 \ldots z_{e_2}\rangle \,, \tag{20}$$

where $\overline{h_2(z_1 \ldots z_{e_2})} := h_2(z_1 \ldots z_k) \oplus 1^{2\lceil \log_2 p \rceil}$. Theorem 3.1 therefore gives us that we can perform a secondary classical search procedure and we can use preprocessing as described in Theorem 3.8 to reduce the cost of the circuit. As $\left|g_2(z_1, \ldots, z_{e_2})\right\rangle$ depends solely upon $z_1 \ldots z_{e_2}$ at all times, after hardcoding is completed, the qubits required to represent it can be removed in addition to the e_2 qubits of the search-space Grover is defined upon. After cancellations of layers of X gates, the 2^k applications of $U_k^\dagger U_\star U_k$ is then simply $2^k + 1$ layers of $2\lceil \log_2 p \rceil$ X gates executed in parallel with 2^k $\wedge_{2\lceil \log_2 p \rceil}(X)$ gates in between each layer.

In relation to the CSSI problem, the security level of SIKE [14] is parameterised by a prime p of the form $2^e 3^f - 1$ where $2^e \approx 3^f$ so that $e \approx p^{1/2}$. The problem of breaking an instance of SIKE-p is then equivalent to finding the unique degree 2^e isogeny defined by the public-parameters of SIKE-p.

Theorem 4.3 (Grover vs CSSI) *Let C_e be the cost (either quantum circuit-size or quantum circuit-depth) of evaluating a degree 2^e isogeny as a reversible quantum circuit. Solving the CSSI problem via Grover's algorithm then has a cost of*

$$O\left(p^{1/4} \cdot {C_e}^{1/2} (\log_2 p)^{1/2}\right). \tag{21}$$

Proof. We can express the asymptotic cost of our attack parameterised by our choice of e_2 (where we recall $e_1 + e_2 = e$) as

$$O\left(p^{1/4} 2^{-e_2/2} \cdot \left[2C_e \cdot \frac{\log_2 p}{\log_2 p} + 2^{e_2} \cdot 2 \log_2 p\right]\right) \tag{22}$$

if we assume $C_{e_1} \leq C_e$ if all other parameters are fixed and the secondary classical-search procedure is in $O\left(2^{e_2} \cdot \log_2 p\right)$ gates as discussed on the previous page. Taking the derivative of (22) gives our optimal value of $e_2 = \log_2\left(\frac{C_e}{\log_2 p}\right)$. □

This is in comparison to simply using the oracle with Grover's algorithm for a circuit-size of $O(p^{1/4} C_e)$. C_e takes $O(e \log_2 e)$ curve operations [10], each of which can be assumed to cost $O\left((\log_2 p)^2 \log_2 \log_2 p\right)$ quantum gates [22, Table 1] (this may be an underestimate). Thus $C_e \in O\left(e(\log_2 e)(\log_2 p)^2(\log_2 \log_2 p)\right)$ and our asymptotic speedup is in $O\left((\log_2 p) \cdot (\log_2 \log_2 p)\right)$. In [16], Grover's algorithm is used to derive estimates on the cost of attacking SIKE for specific security parameters and in Appendix B, we use our result with their methodology.

5 Conclusions

The extent to which the overhead of the quantum oracle can be reduced is clearly an important issue if the cryptographic community is choosing parameters relative to a costing of Grover's algorithm which takes into account both the query-complexity and the cost of the queries themselves. The safest route is of course to simply choose the query-complexity as a lower-bound on the circuit-size for such Grover-based attacks and this protects against our optimisation as we only *increase* the total number of queries.

Our gains have instead been enabled via better use of intermediate computations and exploiting classical computation to create efficient hardcoded circuits, both of which can then be used find an optimal balance between the cost of the quantum oracle and the query-complexity. Whilst our methods are obviously not applicable to all quantum oracles, a cautionary half-way measure between using the lower-bound of query-complexity and the current methodology may be to produce a conservative quantum resource estimate for the cost of the quantum oracle and use the square root of this for the overhead of the quantum oracle when choosing cryptographic parameters relative to Grover's algorithm.

Acknowledgement: This research was supported by funding from EPRSC grant EP/M50645X/1, National Science Foundation grant 183980, National Science Foundation grant 1846166, National Institute of Standards and Technology grant 60NANB17D184, CyberFlorida Collaborative Seed Grant Program and the CyberFlorida Capacity Building Program.

References

[1] John Ahlgren, The Probability Distribution for Draws Until First Success Without Replacement, *arXiv preprint arXiv:1404.1161* (2014).

[2] M. Amy, D. Maslov, M. Mosca and M. Roetteler, A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* **32** (2013), 818–830.

[3] D. Bernstein and B.-Y. Yang, Asymptotically faster quantum algorithms to solve multivariate quadratic equations, in: *International Conference on Post-Quantum Cryptography*, Springer, pp. 487–506, 2018.

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

[5] Denis Xavier Charles, Kristin E. Lauter and Eyal Z. Goren, Cryptographic Hash Functions from Expander Graphs, *J. Cryptology* **22** (2009), 93–113.

[6] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska and Peter Schwabe, *MQDSS—Submission to the NIST post-quantum cryptography project.*, 2017.

[7] Anamaria Costache, Brooke Feigon, Kristin E. Lauter, Maike Massierer and Anna Puskás, Ramanujan graphs in cryptography, *CoRR* **abs/1806.05709** (2018).

[8] Jintai Ding, Ming-Shen Chen, Albrecht Petzoldt, Dieter Schmidt and Bo-Yin Yang, *Gui—Submission to the NIST post-quantum cryptography project. Specification*, 2017.

[9] Jean-Charles Faugere, Kelsey Horan, Delaram Kahrobaei, Marc Kaplan, Elham Kashefi and Ludovic Perret, Fast Quantum Algorithm for Solving Multivariate Quadratic Equations, *arXiv preprint arXiv:1712.07211* (2017).

[10] L. De Feo, D. Jao and J. Plût, Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies, *J. Mathematical Cryptology* **8** (2014), 209–247.

[11] Gray Frank, *Pulse code communication*, March 17 1953, US Patent 2,632,058.

[12] L. Grover, A fast quantum mechanical algorithm for database search, in: *Proc. of the 28th annual ACM symp. on Theory of computing*, ACM, pp. 212–219, 1996.

[13] Andreas Hülsing, Joost Rijneveld, Simona Samardjiska and Peter Schwabe, From 5-pass MQ-based identification to MQ-based signatures., *IACR Cryptology ePrint Archive* **2016** (2016), 708.

[14] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. De Feo, Ba. Hess, A. Jalali, B. Koziel, B. LaMacchia, P. Longa, M. Naehrig, G. Pereira, J. Renes, V. Soukharev and D. Urbanik, *SIKE*, https://sike.org, 2017, Round 2 NIST submission for the standardisation of Post Quantum Crypptography.

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

[16] S. Jaques and J. Schanck, *Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE*, University of Waterloo, Report, 2019, To appear in the proceedings of CRYPTO 2019.

[17] D. Maslov, Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization, *Physical Review A* **93** (2016), 022311.

[18] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, 2010.

[19] A. Petzoldt, M.-S. Chen, J. Ding and B.-Y. Yang, HMFEV- an efficient multivariate signature scheme, in: *International workshop on post-quantum cryptography*, Springer, pp. 205–223, 2017.

[20] A. Petzoldt, M.-S. Chen, B.-Y. Yang, C. Tao and J. Ding, Design principles for HFEv-based multivariate signature schemes, in: *Int. Conference on the Theory and Application of Cryptology and Information Security*, Springer, pp. 311–334, 2015.

[21] Benjamin Pring, Exploiting preprocessing for quantum search to break parameters for MQ cryptosystems, in: *Arithmetic of Finite Fields-7th International Workshop, WAIFI 2018, Revised Selected Papers.*, WAIFI, 2018.

[22] M. Roetteler, M. Naehrig, K. Svore and K. Lauter, Quantum resource estimates for computing elliptic curve discrete logarithms, in: *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, pp. 241–270, 2017.
[23] P. Schwabe and B. Westerbaan, Solving Binary $\mathcal{MQ}$ with Grover's Algorithm, in: *SPACE 2016*, Springer, pp. 303–322, 2016.
[24] P. Selinger, Quantum circuits of T-depth one, *Phys. Rev. A* **87** (2013), 042302.
[25] Seiichiro Tani, An improved claw finding algorithm using quantum walk, in: *International Symposium on Mathematical Foundations of Computer Science*, Springer, pp. 536–547, 2007.
[26] David Urbanik and David Jao, SoK: The problem landscape of SIDH, in: *Proceedings of the 5th ACM on ASIA Public-Key Cryptography Workshop*, ACM, pp. 53–60, 2018.
[27] Paul C Van Oorschot and Michael J Wiener, Parallel collision search with application to hash functions and discrete logarithms, in: *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, ACM, pp. 210–218, 1994.
[28] C. Zalka, Grover's quantum searching algorithm is optimal, *Physical Review A* **60** (1999), 2746.

A Adapting a quantum bit oracle for the $\mathcal{MQ}$ problem over $\mathbb{F}_2$

A quantum bit oracle for the $\mathcal{MQ}$ problem over $\mathbb{F}_2$

In this Section we study how an existing quantum oracle design [23] can be modified to fit under our framework. We first describe the original oracle design [23], how a previous optimisation [21] falls under our framework and how this preexisting method can be improved via our framework to reduce the total circuit-size via use of additional ancilla qubits. We first recall Definition 4.1

Definition 4.1 (The Multivariate Quadratic ($\mathcal{MQ}$) problem over $\mathbb{F}_2$) We define $f^{(1)}(x_1, \ldots, x_n), \ldots, f^{(m)}(x_1, \ldots, x_n) \in \mathbb{F}_2[x_1, \ldots, x_n]$ be m equations of degree two in n variables over the finite field of size 2. The *Multivariate Quadratic* ($\mathcal{MQ}$) problem over $\mathbb{F}_2$ is to find a solution vector $(x_1, \ldots, x_n) \in (\mathbb{F}_2)^n$ such that

$$f^{(1)}(x_1, \ldots, x_n) = \cdots = f^{(m)}(x_1, \ldots, x_n) = 0. \tag{A.1}$$

A quantum bit oracle for the $\mathcal{MQ}$ problem over $\mathbb{F}_2$.

We first describe a quantum bit oracle to solve this problem proposed by Schwabe and Westerbaan [23]. They first perform a classical preprocessing so that 1 is added to each $f^{(k)}(x_1, \ldots, x_n)$. In this way the original system of equations is satisfied when we find an element $x_1 \ldots x_n$ such that $f^{(k)}(x_1, \ldots, x_n) = 1$ for $k = 1, \ldots, m$. Their quantum bit oracle evaluates each multivariate polynomial in a separate register and then uses a single $\wedge_m(X)$ gate to and check if they are satisfied. By noting $x_i x_j = x_j x_i$ and $x_i^2 = x_i$, each multivariate polynomial can be written

$$f^{(k)}(x_1, \ldots, x_n) = \sum_{1 \le i < j \le n} a_{i,j}^{(k)} x_i x_j \oplus \sum_{i=1}^{n} b_i^{(k)} x_i \oplus c^{(k)}, \tag{A.2}$$

where $a_{i,j}^{(k)}, b_i^{(k)}, c^{(k)} \in \mathbb{F}_2$. Schwabe and Westerbaan define the quantum bit oracle as acting upon $n + m + 2$ qubits, so that it uses n qubits for Grover's search-space, m qubits to store the evaluated equations, 1 ancilla qubit to allow the efficient evaluation equations and 1 qubit for the output of the quantum bit oracle.
The evaluation of each $f^{(k)}$ is performed via successively adding the sums

$$x_i \cdot \left(a_{i,i+1}^{(k)} x_{i+1} \oplus \cdots \oplus a_{i,n}^{(k)} x_n \oplus b_i^{(k)} \right) \tag{A.3}$$

onto the m equation registers, one equation at a time. Each step for $i = 1, \ldots, n$ (via an ancilla qubit starting and ending in $|0\rangle$) can be accomplished using at most 1 X gate, $n - i$ $\wedge_1(X)$ gates and a single $\wedge_2(X)$ gate. A single $\wedge_m(X)$ gate is used after all equations are evaluated on the m registers and used to write the output of the quantum bit oracle, which will be 1 if all of the original equations are satisfied.

Applying our framework

A previously published use of preprocessing exploits only a basic form of secondary classical-search (Theorem 3.1) combined with preprocessing (Theorem 3.8), which under our framework can be interpreted by defining U_{n-k} to evaluate m equations of the form

$$f^{(k)}(x_1, \ldots, x_{n-k}) = \sum_{1 \le i < j \le n-k} a_{i,j}^{(k)} x_i x_j \oplus \sum_{i=1}^{n-k} b_i^{(k)} x_i, \tag{A.4}$$

which is possible as they are simply m equations in $n-k$ variables. U_k is then the addition of

$$f^{(k)}(x_1, \ldots, x_{n-k}, z_1, \ldots, z_k) \oplus f^{(k)}(x_1, \ldots, x_{n-k}) \tag{A.5}$$

to each equation register, whilst $U_\star$ is a $\wedge_m(X)$ gate as before. It is easily seen that $C_{\wedge_m(X)}$ is $O(m)$, that the $C_{U_{n-k}}$ is $O(m \cdot (n-k)^2)$ by the discussion on the previous page that C_{U_k} is $O(m \cdot (n-k))$ as hardcoding collapses sums involving $x_i z_j$ to either 0 or x_i and interactions between $z_i z_j$ or z_k to a single bit. The asymptotic cost of Grover's algorithm with the modified quantum bit oracle using secondary classical search and hardcoded bits is therefore

$$O\left(2^{n/2} 2^{-k/2} \cdot \left(m(n-k)^2 + 2^k \cdot m(n-k)\right)\right) \tag{A.6}$$

and by taking the derivative and we find that the optimal $k \approx \log_2(n)$ so that the asymptotic quantum circuit-size of Grover using approximately $n - \log_2 n + m + 2$ qubits and the method described in [21] is $O\left(2^{n/2} \cdot mn^{3/2}\right)$. This asymptotic analysis was not performed in the original paper.

Following a heuristic design pattern with our framework

We use our framework to improve upon this result, obtaining a quantum bit oracle for the $\mathcal{MQ}$ problem over $\mathbb{F}_2$ that uses $n + km + m + 2$ qubits and enables Grover's algorithm to be implemented with a quantum circuit-size of $O\left(2^{n/2} \cdot mn\right)$. This can be done via simply redefining the unitary operators to use Theorem 3.5 in conjunction with Theorem 3.7. By keeping U_{n-k} as before, but defining each unitary $U_{\chi_{n-k+i}}$ for $1 \le i \le k$ by the action of adding only the component

$$z_i \cdot \left(a_{1,i}^{(k)} x_1 \oplus \cdots \oplus a_{n-k,i}^{(k)} x_{n-k} \oplus b_i^{(k)}\right). \tag{A.7}$$

It is clear that these linear sums can be computed and stored on ancilla qubits via Theorem 3.7 and that this cost can be shifted to U_{n-k}. We then have these unitary operator fulfil Theorem 3.5 and that after the shifting of costs to U_{n-k}, we have that U_{χ_i} consists of simple one $\wedge_1(X)$ gate. We can then define $U_\star$ to add the component which only involves the bits $z_1 \ldots z_k$ (which collapse to a hardcoded bit and at most m X gates), execute a $\wedge_m(X)$ gate and uncompute the hardcoded bit again via at most one X gate. In this way the cost for Grover's algorithm (see Theorem 3.5) using this quantum bit oracle becomes

$$O\left(2^{n/2} 2^{-k/2} \left(mn^2 + 2^k m\right)\right), \tag{A.8}$$

hence after optimisation via taking the derivative again with respect to k and simplifying we obtain that the optimal cut to choose is $k \approx \log_2\left(n^2\right)$. This gives us the result that if we allow $n + m(2\log_2 n + 1) + 2$ qubits then we have that Grover's algorithm requires a quantum circuit-size of $O\left(2^{n/2} \cdot mn\right)$.

B Cost estimates of the attacks against SIKE

The authors of [16] consider two cost-metrics which are grounded in real-world concerns. They consider both the total quantum circuit-size (# gates) the algorithm requires and consider a new metric consisting of the product of the quantum circuit-Depth and thethe quantum circuit-Width ($D \times W$). The $D \times W$ metric stems from considering the problems of implementing quantum error-correction and posits that this is a sensible metric as the cost of performing quantum error-correction will be a dominating factor in terms of real-world costs and it must be performed upon qubits which are both idle and being acted upon by quantum gates. Both are important metrics at the current time owing to uncertainty about the eventual architecture of quantum computers.

Figure 1 gives a table of the costs that we have derived using our preprocessing improvements upon Grover-based SIKE attack as given in [16]. The first three rows gives the quantum circuit-complexity for Grover's algorithm and are optimal in terms of both the circuit-size and the $D \times W$ metric, whilst the last four rows give both the optimal circuit-size and $D \times W$ versions of both Tani's algorithm [25] and the van Oorschot-Wiener approach [27]. We do not examine the issue of constraints as in [16], but note that our comments about parallelism strategies may allows gains in this area.

	SIKE-434			SIKE-610		
Attack cost	G	D	W	G	D	W
Grover [16]	132	122	10	177	167	10
Grover (Ours with assumptions from [16])	126	116	10	171	160	10
Grover (Ours with higher costs)	130	120	10	175	165	10
Tani[16] (optimal # gates)	124	114	25	169	159	25
Tani[16] (optimal $D \times W$)	131	122	10	177	166	10
VW [16] (optimal # gates)	132	14	128	177	14	173
VW [16] (optimal $D \times W$)	132	14	128	177	14	173

Figure 1: Comparison between conservative estimations (in $\log_2$) for quantum-circuit-complexity (Gates, Depth, Width) in $\log_2$ required for various approaches to cryptanalysis of SIKE-p, including the proposed Depth×Width cost-metric [16].

The first row details the quantum circuit-complexity of Grover from [16] using the assumption that the cost of the quantum oracle is derived from computing one degree-$2^{e/2}$ isogeny for a cost of $e/2 \log_2(e/2)$ elliptic curve operations and that these elliptic curve operations cost $4 \log_2 p \log_2 \log_2 p$ quantum gates, which they state is a conservative estimate and hence useful to derive security estimates from.

The second row uses our optimisation, but with the cost of computing our degree-2^{e_1} isogeny as $e \log_2 e$ elliptic curve operations (recall $e_1 + e_2 = e$) and assumes these elliptic curve operations again cost $4 \log_2 p \log_2 \log_2 p$ quantum gates.

The third row uses our optimisation and the assumption that the degree-2^{e_1} isogeny costs $e \log_2 e$ elliptic curve operations but assumes these curve operations cost $4(\log_2 p)^2 \log_2 \log_2 p$ quantum gates. This estimate, whilst perhaps still conservative is perhaps more realistic [22]. We note that even though we have increased the costs, our optimisation still has a lower quantum-circuit complexity and note that row 2 implies that Grover may be comparable with Tani's algorithm in the gate-based metric and has the potential to beat Tani's algorithm in the $D \times W$ metric. This stems from the fact that even though we are assuming higher individual cost components (row 3), the algorithmic advantages are such that we have a $O(\log_2 \log_2 p)$ advantage in circuit-size over that described in row 1 from [16].

- Grover may be superior in the Depth × Width-cost metric. For SIKE-434 we have a cost of 2^{126} for Grover's algorithm compared to 2^{132} for Tani's algorithm and for SIKE-610, the cost is 2^{170} compared to Tani's cost of 2^{176}.
- Grover may be competitive in the gate based metric. For SIKE-434 this translates into a cost of 2^{126} for Grover's algorithm compared to 2^{124} for Tani's algorithm and for SIKE-610, a cost of 2^{171} compared to Tani's cost of 2^{169}.