On Quantum Chosen-Ciphertext Attacks and Learning with Errors

Gorjan Alagic
QuICS, University of Maryland, MD, USA
NIST, Gaithersburg, MD, USA
galagic@umd.edu

Stacey Jeffery
QuSoft, Amsterdam, Netherlands
CWI, Amsterdam, Netherlands
smjeffery@gmail.com

Maris Ozols
QuSoft, Amsterdam, Netherlands
University of Amsterdam, Netherlands
marozols@gmail.com

Alexander Poremba
Computing and Mathematical Sciences, Caltech, Pasadena, CA, USA
aporemba@caltech.edu

Abstract

Quantum computing is a significant threat to classical public-key cryptography. In strong "quantum access" security models, numerous symmetric-key cryptosystems are also vulnerable. We consider classical encryption in a model which grants the adversary quantum oracle access to encryption and decryption, but where the latter is restricted to non-adaptive (i.e., pre-challenge) queries only. We define this model formally using appropriate notions of ciphertext indistinguishability and semantic security (which are equivalent by standard arguments) and call it QCCA1 in analogy to the classical CCA1 security model. Using a bound on quantum random-access codes, we show that the standard PRF-based encryption schemes are QCCA1-secure when instantiated with quantum-secure primitives.

We then revisit standard IND-CPA-secure Learning with Errors (LWE) encryption and show that leaking just one quantum decryption query (and no other queries or leakage of any kind) allows the adversary to recover the full secret key with constant success probability. In the classical setting, by contrast, recovering the key requires a linear number of decryption queries. The algorithm at the core of our attack is a (large-modulus version of) the well-known Bernstein-Vazirani algorithm. We emphasize that our results should **not** be interpreted as a weakness of these cryptosystems in their stated security setting (i.e., post-quantum chosen-plaintext secrecy). Rather, our results mean that, if these cryptosystems are exposed to chosen-ciphertext attacks (e.g., as a result of deployment in an inappropriate real-world setting) then quantum attacks are even more devastating than classical ones.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Security and privacy → Cryptanalysis and other attacks

Keywords and phrases quantum chosen-ciphertext security, quantum attacks, learning with errors

Digital Object Identifier 10.4230/LIPIcs.TQC.2019.1

Related Version A full version of the paper is available at `https://eprint.iacr.org/2018/1185`.

Funding *Gorjan Alagic*: Supported by National Science Foundation (NSF) grant CCF-1763736.
Stacey Jeffery: Supported by an NWO WISE Grant and NWO Veni Innovational Research Grant under project number 639.021.752.
Maris Ozols: Supported by Leverhulme Trust Early Career Fellowship (ECF-2015-256).
Alexander Poremba: Partial support from AFOSR YIP award number FA9550-16-1-0495, the IQIM – an NSF Physics Frontiers Center (NSF Grant PHY- 1733907), and the Kortschak Scholars program.

Acknowledgements We thank Ronald de Wolf for helpful discussions and Jop Briët for Lemma 25.

© Gorjan Alagic, Stacey Jeffery, Maris Ozols, and Alexander Poremba;
licensed under Creative Commons License CC-BY

1 Introduction

Large-scale quantum computers pose a dramatic threat to classical cryptography. The ability of such devices to run Shor's efficient quantum factoring algorithm (and its variants) would lead to devastation of the currently deployed public-key cryptography infrastructure [8, 25]. This threat has led to significant work on so-called "post-quantum" alternatives, where a prominent category is occupied by cryptosystems based on the *Learning with Errors* (LWE) problem of solving noisy linear equations over $\mathbb{Z}_q$ [23] and its variants [8, 22].

In addition to motivating significant work on post-quantum cryptosystems, the threat of quantum computers has spurred general research on secure classical cryptography in the presence of quantum adversaries. One area in particular explores security models where a quantum adversary gains quantum control over portions of a classical cryptosystem. In such models, a number of basic symmetric-key primitives can be broken by simple quantum attacks based on Simon's algorithm [16, 17, 15, 24, 26]. It is unclear if the assumption behind these models is plausible for typical physical implementations of symmetric-key cryptography. However, attacks that involve quantumly querying a classical function are always available in scenarios where the adversary has access to a circuit for the relevant function. This is the case for hashing, public-key encryption, and circuit obfuscation. Moreover, understanding this model is crucial for gauging the degree to which any physical cryptographic device must be resistant to reverse engineering or forced quantum behavior (consider the so-called "frozen smart card" example [10]). For instance, one may reasonably ask: *what happens to the security of a classical cryptosystem when the device leaks only a single quantum query to the adversary?*

When deciding which functions the adversary might have (quantum) access to, it is worth recalling the classical setting. For classical symmetric-key encryption, a standard approach considers the security of cryptosystems when exposed to so-called chosen-plaintext attacks (CPA). This notion encompasses all attacks in which an adversary attempts to defeat security (by, e.g., distinguishing ciphertexts or extracting key information) using oracle access to the function which encrypts plaintexts with the secret key. This approach has been highly successful in developing cryptosystems secure against a wide range of realistic real-world attacks. An analogous class, the so-called chosen-ciphertext attacks (CCA), are attacks in which the adversary can make use of oracle access to decryption. For example, a well-known attack due to Bleichenbacher [4] only requires access to an oracle that decides if the input ciphertext is encrypted according to a particular RSA standard. We will consider analogues of both CPA and CCA attacks, in which the relevant functions are quantumly accessible to the adversary.

Prior works have formalized the quantum-accessible model for classical cryptography in several settings, including unforgeable message authentication codes and digital signatures [6, 5], encryption secure against quantum chosen-plaintext attacks (QCPA) [7, 10], and encryption secure against *adaptive* quantum chosen-ciphertext attacks (QCCA2) [6].

1.1 Our Contributions

1.1.1 The model

In this work, we define a quantum-secure model of encryption called QCCA1. This model grants *non-adaptive* access to the decryption oracle, and is thus intermediate between QCPA and QCCA2. Studying weaker and intermediate models is a standard and useful practice in cryptography. In fact, CPA and CCA2 are intermediate models themselves, being strictly weaker than authenticated encryption. Our particular intermediate model is naturally motivated: it is sufficent for a new and interesting quantum attack on LWE encryption.

As is typical, the "challenge" in QCCA1 can be semantic, or take the form of an indistinguishability test. This leads to natural security notions for symmetric-key encryption, which we call IND-QCCA1 and SEM-QCCA1, respectively. Following previous works, it is straightforward to prove that IND-QCCA1 and SEM-QCCA1 are equivalent [7, 10, 6].

We prove IND-QCCA1 security for two symmetric-key encryption schemes, based on standard assumptions. Specifically, we show that the standard encryption scheme based on quantum-secure pseudorandom functions (QPRF) is IND-QCCA1-secure. We remark that QPRFs can be constructed from quantum-secure one-way functions [28]. Our security proofs use a novel technique, in which we control the amount of information that the adversary can extract from the oracles and store in their internal quantum state (prior to the challenge) by means of a certain bound on quantum random-access codes.

1.1.2 A quantum-query attack on LWE

We then revisit the aforementioned question: what happens to a post-quantum cryptosystem if it leaks a single quantum query? Our main result is that standard IND-CPA-secure LWE-based encryption schemes can be completely broken using only *a single quantum decryption query* and no other queries or leakage of any kind. In our attack, the adversary recovers the complete secret key with constant success probability. In standard bit-by-bit LWE encryption, a single classical decryption query can yield at most one bit of the secret key; the classical analogue of our attack thus requires $n \log q$ queries. The attack is essentially an application of a modulo-q variant of the Bernstein-Vazirani algorithm [3]. Our new analysis shows that this algorithm correctly recovers the key with constant success probability, despite the decryption function only returning an inner product which is rounded to one of two values. We show that the attack applies to four variants of standard IND-CPA-secure LWE-based encryption: the symmetric-key and public-key systems originally described by Regev [23], the FrodoPKE scheme[1] [18, 1], and standard Ring-LWE [19, 20].

1.1.3 Important caveats

Our results challenge the idea that LWE is unconditionally "just as secure" quantumly as it is classically. Nonetheless, the reader is cautioned to interpret our work carefully. Our results do *not* indicate a weakness in LWE (or any LWE-based cryptosystem) in the standard post-quantum security model. Since it is widely believed that quantum-algorithmic attacks will need to be launched over purely classical channels, post-quantum security does not allow for quantum queries to encryption or decryption oracles. Moreover, while our attack does offer a dramatic quantum speedup (i.e., one query vs. linear queries), the classical attack is already efficient. The schemes we attack are already insecure in the classical chosen-ciphertext setting, but can be modified to achieve chosen-ciphertext security [9].

[1] FrodoPKE is an IND-CPA-secure building block in the IND-CCA2-secure post-quantum cryptosystem "FrodoKEM" [1]. Our results do not affect the post-quantum security of Frodo and do not contradict the CCA2 security of FrodoKEM.

1.1.4 Related work

We remark that Grilo, Kerenidis and Zijlstra recently observed that a version of LWE with so-called "quantum samples" can be solved efficiently (as a learning problem) using Bernstein-Vazirani [14]. Our result, by contrast, demonstrates an actual cryptographic attack on standard cryptosystems based on LWE, in a plausible security setting. Moreover, in terms of solving the learning problem, our analysis shows that constant success probability is achievable with only a single query, whereas [14] require a number of queries which is at least linear in the modulus q. In particular, our cryptographic attack succeeds with a single query even for superpolynomial modulus.

1.2 Technical summary of results

1.2.1 Security model and basic definitions

We set down the basic QCCA1 security model, adapting the ideas of [5, 10]. An encryption scheme is a triple $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of algorithms (key generation, encryption, and decryption, respectively) satisfying $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$ for any key $k \leftarrow \mathsf{KeyGen}$ and message m. In what follows, all oracles are quantum, meaning a function f is accessed by the unitary $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$. We define indistinguishability and semantic security as follows.

▶ **Definition 1** (informal). *Π is IND-QCCA1 if no quantum polynomial-time algorithm (QPT) $\mathcal{A}$ can succeed at the following experiment with probability better than $1/2 + \mathrm{negl}(n)$.*

1. *A key $k \leftarrow \mathsf{KeyGen}(1^n)$ and a uniformly random bit $b \xleftarrow{\$} \{0, 1\}$ are generated; $\mathcal{A}$ gets access to oracles Enc_k and Dec_k, and outputs (m_0, m_1);*
2. *$\mathcal{A}$ gets $\mathsf{Enc}_k(m_b)$ and access to an oracle for Enc_k, and outputs a bit b'; $\mathcal{A}$ wins if $b = b'$.*

▶ **Definition 2** (informal). *Consider the following game with a QPT $\mathcal{A}$.*

1. *A key $k \leftarrow \mathsf{KeyGen}(1^n)$ is generated; $\mathcal{A}$ gets access to oracles Enc_k, Dec_k and outputs circuits (Samp, h, f);*
2. *Sample $m \leftarrow \mathsf{Samp}$; $\mathcal{A}$ receives $h(m)$, $\mathsf{Enc}_k(m)$, and access to an oracle for Enc_k only, and outputs a string s; $\mathcal{A}$ wins if $s = f(m)$.*

Then Π is SEM-QCCA1 if for every QPT $\mathcal{A}$ there exists a QPT $\mathcal{S}$ with the same winning probability but which does not get $\mathsf{Enc}_k(m)$ in step 2.

▶ **Theorem 3.** *A symmetric-key encryption scheme is IND-QCCA1 if and only if it is SEM-QCCA1.*

1.2.2 Secure constructions

Next, we show that standard pseudorandom-function-based encryption is QCCA1-secure, provided that the underlying PRF is quantum-secure (i.e., is a QPRF.) A QPRF can be constructed from any quantum-secure one-way function, or directly from the LWE assumption [28]. Given a PRF $f = \{f_k\}_k$, define $\mathsf{PRFscheme}[f]$ to be the scheme which encrypts a plaintext m using randomness r via $\mathsf{Enc}_k(m; r) = (r, f_k(r) \oplus m)$ and decrypts in the obvious way.

▶ **Theorem 4.** *If f is a QPRF, then $\mathsf{PRFscheme}[f]$ is IND-QCCA1-secure.*

In the full version of this work, we also analyze a standard permutation-based scheme. Quantum-secure PRPs (i.e., QPRPs) can be obtained from quantum-secure one-way functions [29]. Given a PRP $P = \{P_k\}_k$, define $\mathsf{PRPscheme}[P]$ to be the scheme that encrypts a plaintext m using randomness r via $\mathsf{Enc}_k(m;r) = P_k(m||r)$, where $||$ denotes concatenation; to decrypt, apply P_k^{-1} and discard r.

▶ **Theorem 5.** *If P is a* QPRP*, then* PRPscheme$[P]$ *is* IND-QCCA1*-secure.*

We briefly describe our proof techniques for Theorems 4 and 5. In the indistinguishability game, the adversary can use the decryption oracle prior to the challenge to (quantumly) encode information about the relevant pseudorandom function instance (i.e., f_k or P_k) in his private, poly-sized quantum memory. To establish security, it is enough to show that this encoded information cannot help the adversary compute the value of the relevant function at the particular randomness used in the challenge. To prove this, we use a bound on quantum random access codes (QRAC), where, informally, a QRAC is a mapping from N-bit strings x to d-dimensional states ϱ_x, such that given ϱ_x, and any $j \in [N]$, x_j can be recovered with some probability.

▶ **Lemma 6.** *The average bias of a quantum random access code with shared randomness that encodes N bits into a d-dimensional quantum state is $O(\sqrt{N^{-1}\log d})$. In particular, if $N = 2^n$ and $d = 2^{\mathrm{poly}(n)}$ the bias is $O(2^{-n/2}\,\mathrm{poly}(n))$.*

1.2.3 Key recovery against LWE

Our attack on LWE encryption uses a new analysis of a large-modulus variant of the Bernstein-Vazirani algorithm [3], in the presence of a certain type of "rounding" noise.

1.2.4 Quantum algorithm for linear rounding functions

Given integers $n \geq 1$ and $q \geq 2$, define a keyed family of (binary) linear rounding functions, $\mathsf{LRF}_{\boldsymbol{k},q} : \mathbb{Z}_q^n \longrightarrow \{0,1\}$, with key $\boldsymbol{k} \in \mathbb{Z}_q^n$, as follows:

$$\mathsf{LRF}_{\boldsymbol{k},q}(\boldsymbol{x}) := \begin{cases} 0 & \text{if } |\langle \boldsymbol{x}, \boldsymbol{k}\rangle| \leq \lfloor \frac{q}{4} \rfloor, \\ 1 & \text{otherwise.} \end{cases}$$

Here $\langle \cdot, \cdot \rangle$ denote the inner product modulo q. Our main technical contribution is the following.

▶ **Theorem 7** (informal)**.** *There exists a quantum algorithm that runs in time $O(n)$, makes one quantum query to $\mathsf{LRF}_{\boldsymbol{k},q}$ (with unknown $\boldsymbol{k} \in \mathbb{Z}_q^n$), and outputs $\boldsymbol{k}$ with probability $4/\pi^2 - O(1/q)$.*

We also show that the same algorithm succeeds against more generalized function classes, in which the oracle indicates which "segment" of $\mathbb{Z}_q$ the exact inner product belongs to.

1.2.5 One quantum query against LWE

Finally, we revisit our central question of interest: what happens to a post-quantum cryptosystem if it leaks a single quantum query? We show that, in standard LWE-based schemes, the decryption function can (with some simple modifications) be viewed as a special case of a linear rounding function, as above. In standard symmetric-key or public-key LWE, for instance, we decrypt a ciphertext $(\boldsymbol{a}, c) \in \mathbb{Z}_q^{n+1}$ with key $\boldsymbol{k}$ by outputting 0 if $|c - \langle \boldsymbol{a}, \boldsymbol{k}\rangle| \leq \left\lfloor \frac{q}{4} \right\rfloor$ and 1 otherwise. In standard Ring-LWE, we decrypt a ciphertext (u, v) with key k (here u, v, k are polynomials in $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$) by outputting 0 if the constant coefficient of $v - k \cdot u$ is small, and 1 otherwise.

Each of these schemes is secure against adversaries with classical encryption oracle access, under the LWE assumption. If adversaries also gain classical decryption access, then it's not hard to see that a linear number of queries is necessary and sufficient to recover the private key. Our main result is that, by contrast, only a *single* quantum decryption query is required to achieve this total break. Indeed, in all three constructions described above, one can use the decryption oracle to build an associated oracle for a linear rounding function which hides the secret key. The following can then be shown using Theorem 7.

▶ **Theorem 8** (informal). *Let Π be standard* LWE *or standard* Ring-LWE *encryption (either symmetric-key, or public-key.) Let n be the security parameter. Then there is an efficient quantum algorithm that runs in time $O(n)$, uses one quantum query to the decryption function* $\mathsf{Dec}_{\boldsymbol{k}}$ *of Π,and outputs the secret key with constant probability.*

It's natural to ask whether a similar quantum speedup can be achieved using encryption queries. In the full version of this article, we show that this is possible in a model in which the adversary is allowed to select some of the random coins used to encrypt.

1.3 Organization

The remainder of this paper is organized as follows. In Section 2, we outline preliminary ideas that we will make use of, including cryptographic concepts, and notions from quantum algorithms. In Section 3, we define the QCCA1 model, including the two equivalent versions IND-QCCA1 and SEM-QCCA1. In Section 4, we define the PRF scheme, and show that they are IND-QCCA1-secure. In Section 5, we show how a generalization of the Bernstein-Vazirani algorithm works with probability bounded from below by a constant, even when the oracle outputs rounded values. In Section 6, we use the results of Section 5 to prove that a single quantum decryption query is enough to recover the secret key in various versions of LWE-encryption.

2 Preliminaries

2.1 Basic notation and conventions

Selecting an element x uniformly at random from a finite set X will be written as $x \xleftarrow{\$} X$. If we are generating a vector or matrix with entries in $\mathbb{Z}_q$ by sampling each entry independently according to a distribution χ on $\mathbb{Z}_q$, we will write, e.g., $\boldsymbol{v} \xleftarrow{\chi} \mathbb{Z}_q^n$. Given a matrix A, A^T will denote the transpose of A. We will view elements $\boldsymbol{v}$ of $\mathbb{Z}_q^n$ as column vectors; the notation $\boldsymbol{v}^\mathsf{T}$ then denotes the corresponding row vector. The notation $\mathrm{negl}(n)$ denotes some function of n which is smaller than every inverse-polynomial. We denote the concatenation of strings x and y by $x||y$. We abbreviate classical probabilistic polynomial-time algorithms as PPT algorithms. By *quantum algorithm* (or QPT) we mean a polynomial-time uniform family of quantum circuits, where each circuit in the family is described by a sequence of unitary gates and measurements. In general, such an algorithm may receive (mixed) quantum states as inputs and produce (mixed) quantum states as outputs. Sometimes we will restrict QPTs implicitly; for example, if we write $\Pr[\mathcal{A}(1^n) = 1]$ for a QPT $\mathcal{A}$, it is implicit that we are only considering those QPTs that output a single classical bit.

A function $f : \{0,1\}^m \to \{0,1\}^\ell$ defines a unitary operator $U_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$ on $m + \ell$ qubits where $x \in \{0,1\}^m$ and $y \in \{0,1\}^\ell$. When we say that a quantum algorithm $\mathcal{A}$ gets (adaptive) quantum oracle access to f (written $\mathcal{A}^f$), we mean that $\mathcal{A}$ can apply U_f. Recall that a symmetric-key encryption scheme is a triple of classical probabilistic algorithms

$(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ whose run-times are polynomial in some security parameter n. Such a scheme must satisfy the following property: when a key k is sampled by running $\mathsf{KeyGen}(1^n)$, then it holds that $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$ for all m except with negligible probability in n. In this work, all encryption schemes will be fixed-length, i.e., the length of the message m will be a fixed (at most polynomial) function of n.

Since the security notions we study are unachievable in the information-theoretic setting, all adversaries will be modeled by QPTs. When security experiments require multiple rounds of interaction with the adversary, $\mathcal{A}$ is implicitly split into multiple QPTs (one for each round), and each algorithm forwards its internal (quantum) state to the next algorithm in the sequence.

2.2 Quantum-secure pseudorandomness

Let $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^\ell$ be an efficiently computable function, where n, m, ℓ are integers and where f defines a family of functions $\{f_k\}_{k \in \{0,1\}^n}$ with $f_k(x) = f(k, x)$. We say f is a *quantum-secure pseudorandom function* (or QPRF) if, for every QPT $\mathcal{A}$,

$$\left| \Pr_{k \xleftarrow{\$} \{0,1\}^n} \left[\mathcal{A}^{f_k}(1^n) = 1\right] - \Pr_{g \xleftarrow{\$} \mathcal{F}_m^\ell} \left[\mathcal{A}^{g}(1^n) = 1\right] \right| \leq \mathrm{negl}(n)\,. \tag{1}$$

Here $\mathcal{F}_m^\ell$ denotes the set of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$. The standard method for constructing a pseudorandom function from a one-way function produces a QPRF, provided that the one-way function is quantum-secure [13, 12, 28].

2.3 Quantum random access codes

A *quantum random access code* (QRAC) is a two-party scheme for the following scenario involving two parties Alice and Bob [21]:

1. Alice gets $x \in \{0,1\}^N$ and encodes it as a d-dimensional quantum state ϱ_x.
2. Bob receives ϱ_x from Alice, and some index $i \in \{1, \ldots, N\}$, and is asked to recover the i-th bit of x, by performing some measurement on ϱ_x.
3. They win if Bob's output agrees with x_i and lose otherwise.

We can view a QRAC scheme as a pair of (not necessarily efficient) quantum algorithms: one for encoding, and another for decoding. We remark that the definition of a QRAC does not bound on the size of ϱ_x; the interesting question is with what parameters a QRAC can actually exist.

A variation of the above scenario allows Alice and Bob to use *shared randomness* in their encoding and decoding operations [2]. Hence, Alice and Bob can pursue probabilistic strategies with access to the same random variable.

Define the average bias of a QRAC with shared randomness as $\epsilon = p_{\mathrm{win}} - 1/2$, where p_{win} is the winning probability averaged over $x \xleftarrow{\$} \{0,1\}^N$ and $i \xleftarrow{\$} \{1, \ldots, N\}$.

3 The QCCA1 security model

3.1 Quantum oracles

In our setting, adversaries will (at various times) have quantum oracle access to the classical functions Enc_k and Dec_k. The case of the deterministic decryption function Dec_k is simple: the adversary gets access to the unitary operator $U_{\mathsf{Dec}_k} : |c\rangle|m\rangle \mapsto |c\rangle|m \oplus \mathsf{Dec}_k(c)\rangle$. For encryption, to satisfy IND-CPA security, Enc_k must be probabilistic and thus does not correspond to any single unitary operator. Instead, each encryption oracle call of the adversary will be answered by applying a unitary sampled uniformly from the family $\{U_{\mathsf{Enc}_k,r}\}_r$ where

$$U_{\mathsf{Enc}_k,r} : |m\rangle|c\rangle \mapsto |m\rangle|c \oplus \mathsf{Enc}_k(m;r)\rangle$$

and r varies over all possible values of the randomness register of Enc_k. Note that, since Enc_k and Dec_k are required to be probabilistic polynomial-time algorithms provided by the underlying classical symmetric-key encryption scheme, both $U_{\mathsf{Enc}_k,r}$ and U_{Dec_k} correspond to efficient and reversible quantum operations. For the sake of brevity, we adopt the convenient notation Enc_k and Dec_k to refer to the above quantum oracles for encryption and decryption respectively.

3.2 Ciphertext indistinguishability

We now define indistinguishability of encryptions (for classical, symmetric-key schemes) against non-adaptive quantum chosen-ciphertext attacks.

▶ **Definition 9** (IND-QCCA1). *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme,* $\mathcal{A}$ *a* QPT*, and* n *the security parameter. Define* $\mathsf{IndGame}(\Pi, \mathcal{A}, n)$ *as follows.*

1. Setup: *A key* $k \leftarrow \mathsf{KeyGen}(1^n)$ *and a bit* $b \xleftarrow{\$} \{0,1\}$ *are generated;*
2. Pre-challenge: $\mathcal{A}$ *gets access to oracles* Enc_k *and* Dec_k*, and outputs* (m_0, m_1)*;*
3. Challenge: $\mathcal{A}$ *gets* $\mathsf{Enc}_k(m_b)$ *and access to* Enc_k *only, and outputs a bit* b'*;*
4. Resolution: $\mathcal{A}$ *wins if* $b = b'$.

Then Π *has indistinguishable encryptions under non-adaptive quantum chosen ciphertext attack (or is* IND-QCCA1*) if, for every* QPT $\mathcal{A}$,

$$\Pr[\mathcal{A} \textit{ wins } \mathsf{IndGame}(\Pi, \mathcal{A}, n)] \leq 1/2 + \mathrm{negl}(n)\,.$$

By inspection, one immediately sees that our definition lies between the established notions of IND-QCPA and IND-QCCA2 [7, 10, 6]. It will later be convenient to work with a variant of the game IndGame, which we now define.

▶ **Definition 10** (IndGame′). *We define the experiment* $\mathsf{IndGame}'(\Pi, \mathcal{A}, n)$ *as* $\mathsf{IndGame}(\Pi, \mathcal{A}, n)$*, except that in the pre-challenge phase* $\mathcal{A}$ *only outputs a single message* m*, and in the challenge phase* $\mathcal{A}$ *receives* $\mathsf{Enc}_k(m)$ *if* $b = 0$*, and* $\mathsf{Enc}_k(x)$ *for a uniformly random message* x *if* $b = 1$.

Working with IndGame′ rather than IndGame does not change security. Specifically (as we show in Appendix B), Π is IND-QCCA1 if and only if, for every QPT $\mathcal{A}$,

$$\Pr[\mathcal{A} \text{ wins } \mathsf{IndGame}'(\Pi, \mathcal{A}, n)] \leq 1/2 + \mathrm{negl}(n)\,.$$

3.3 Semantic security

In semantic security, rather than choosing a pair of challenge plaintexts, the adversary chooses a *challenge template*: a triple of circuits (Samp, h, f), where Samp outputs plaintexts from some distribution $\mathcal{D}_{\mathsf{Samp}}$, and h and f are functions with domain the support of $\mathcal{D}_{\mathsf{Samp}}$. The intuition is that Samp is a distribution of plaintexts m for which the adversary, if given information $h(m)$ about m together with an encryption of m, can produce some new information $f(m)$.

▶ **Definition 11** (SEM-QCCA1). *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme, and consider the following experiment,* $\mathsf{SemGame}(b)$*, (with parameter* $b \in \{\mathsf{real}, \mathsf{sim}\}$*) with a* QPT $\mathcal{A}$*:*

1. Setup: *A key* $k \leftarrow \mathsf{KeyGen}(1^n)$ *is generated;*
2. Pre-challenge: *$\mathcal{A}$ gets access to oracles* Enc_k *and* Dec_k*, and outputs a challenge template* (Samp, h, f)*;*
3. Challenge: *A plaintext* $m \xleftarrow{\$} \mathsf{Samp}$ *is generated; $\mathcal{A}$ receives $h(m)$ and gets access to an oracle for* Enc_k *only; if* $b = \mathsf{real}$*, $\mathcal{A}$ also receives* $\mathsf{Enc}_k(m)$*; $\mathcal{A}$ outputs a string s;*
4. Resolution: *$\mathcal{A}$ wins if $s = f(m)$.*

We say Π has semantic security under non-adaptive quantum chosen-ciphertext attack (or is SEM-QCCA1*) if, for every* QPT *$\mathcal{A}$, there exists a* QPT *$\mathcal{S}$ such that the challenge templates output by $\mathcal{A}$ and $\mathcal{S}$ are identically distributed, and*

$$\left|\Pr[\mathcal{A} \textit{ wins } \mathsf{SemGame}(\mathsf{real})] - \Pr[\mathcal{S} \textit{ wins } \mathsf{SemGame}(\mathsf{sim})]\right| \leq \mathrm{negl}(n)\,.$$

Our definition is a straightforward modification of SEM-QCPA [10, 6]; the modification is to give $\mathcal{A}$ and $\mathcal{S}$ oracle access to Dec_k in the pre-challenge phase.

▶ **Theorem 12.** *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric-key encryption scheme. Then, Π is* IND-QCCA1*-secure if and only if Π is* SEM-QCCA1*-secure.*

The classical proof of the above (see, e.g., [11]) carries over to the quantum case. This was already observed for the case of QCPA by [10], and extends easily to the case where both the adversary and the simulator gain oracle access to Dec_k in the pre-challenge phase.[2]

4 Secure Constructions

4.1 PRF scheme

Let us first recall the standard symmetric-key encryption based on pseudorandom functions.

▶ **Construction 13** (PRF scheme). *Let n be the security parameter and let $f : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be an efficient family of functions $\{f_k\}_k$. Then, the symmetric-key encryption scheme* $\mathsf{PRFscheme}[f] = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is defined as follows:*

1. KeyGen: *output* $k \xleftarrow{\$} \{0,1\}^n$*;*
2. Enc: *to encrypt* $m \in \{0,1\}^n$*, choose* $r \xleftarrow{\$} \{0,1\}^n$ *and output* $(r, f_k(r) \oplus m)$*;*
3. Dec: *to decrypt* $(r, c) \in \{0,1\}^n \times \{0,1\}^n$*, output* $c \oplus f_k(r)$*;*

We chose a simple set of parameters for the PRF, so that key length, input size, and output size are all equal to the security parameter. It is straightforward to check that the definition (and our results) are valid for any polynomial-size parameter choices. We show that the above scheme satisfies QCCA1, provided that the underlying PRF is secure against quantum queries.

▶ **Theorem 14.** *If f is a* QPRF*, then* $\mathsf{PRFscheme}[f]$ *is* IND-QCCA1*-secure.*

Proof. Fix a QPT adversary $\mathcal{A}$, split into pre-challenge algorithm $\mathcal{A}_1$ and challenge algorithm $\mathcal{A}_2$, against $\Pi := \mathsf{PRFscheme}[f] = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. Let n denote the security parameter.

We will work with the single-message variant of IndGame, IndGame′, described below as Game 0. In Appendix B, we show that Π is IND-QCCA1 if and only if no QPT adversary can win IndGame′ with non-negligible bias. We first show that a version of IndGame′ where we

[2] In fact, the proof works even if Dec_k access is maintained during the challenge, so the result is really that IND-QCCA2 is equivalent to SEM-QCCA2.

replace f with a random function, called GAME 1 below, is indistinguishable from $\mathsf{IndGame}'$, so that the winning probabilities cannot differ by a non-negligible amount. We then prove that no adversary can win GAME 1 with non-negligible bias by showing how any adversary for GAME 1 can be used to make a quantum random access code with the same bias.

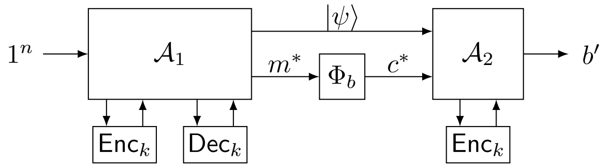

Figure 1 $\mathsf{IndGame}'$ from Definition 10.

Game 0. This is the game $\mathsf{IndGame}'(\Pi, \mathcal{A}, n)$, which we briefly review for convenience (see also Figure 1). In the pre-challenge phase, $\mathcal{A}_1$ gets access to oracles Enc_k and Dec_k, and outputs a message m^* while keeping a private state $|\psi\rangle$ for the challenge phase. In the challenge phase, a random bit $b \xleftarrow{\$} \{0,1\}$ is sampled, and $\mathcal{A}_2$ is run on input $|\psi\rangle$ and a challenge ciphertext

$$c^* := \Phi_b(m^*) := \begin{cases} \mathsf{Enc}_k(m^*) & \text{if } b = 0, \\ \mathsf{Enc}_k(x) & \text{if } b = 1. \end{cases}$$

Here $\mathsf{Enc}_k(x) := (r^*, f_k(r^*) \oplus x)$ where r^* and x are sampled uniformly at random. In the challenge phase, $\mathcal{A}_2$ only has access to Enc_k and must output a bit b'. $\mathcal{A}$ wins if $\delta_{bb'} = 1$, so we call $\delta_{bb'}$ the outcome of the game.

Game 1. This is the same game as GAME 0, except we replace f_k with a uniformly random function $F : \{0,1\}^n \to \{0,1\}^n$.

First, we show that for any adversary $\mathcal{A}$, the outcome when $\mathcal{A}$ plays GAME 0 is at most negligibly different from the outcome when $\mathcal{A}$ plays GAME 1. We do this by constructing a quantum distinguisher $\mathcal{D}$ that distinguishes QPRF $\{f_k\}_k$ from a true random function, with advantage $\big|\Pr[1 \leftarrow \text{GAME } 0] - \Pr[1 \leftarrow \text{GAME } 1]\big|$, which must then be negligible since f is a QPRF. The distinguisher $\mathcal{D}$ gets quantum oracle access to a function g, which is either f_k, for a random k, or a random function, and proceeds by simulating $\mathcal{A}$ playing $\mathsf{IndGame}'$ as follows:

1. Run $\mathcal{A}_1$, answering encryption queries using calls to g instead of f_k, and decryption queries using quantum oracle calls to g: $|r\rangle|c\rangle|m\rangle \mapsto |r\rangle|c\rangle|m \oplus c\rangle \mapsto |r\rangle|c\rangle|m \oplus c \oplus g(r)\rangle$;
2. Simulate the challenge phase by sampling $b \xleftarrow{\$} \{0,1\}$ and encrypting the challenge using g in place of f_k; run $\mathcal{A}_2$ and simulate encryption queries as before;
3. When $\mathcal{A}_2$ outputs b', output $\delta_{bb'}$.

To show that no QPT adversary can win GAME 1 with non-negligible probability, we design a QRAC from any adversary, and use the lower bound on the bias given in Lemma 6.

Intuition: In an encryption query, $\mathcal{A}_1$ or $\mathcal{A}_2$, queries a message, or superposition of messages $\sum_m |m\rangle$, and gets back $\sum_m |m\rangle|r, m \oplus F(r)\rangle$ for a random r, from which he can easily obtain $(r, F(r))$. So in essence, an encryption query is just classically sampling a random point of F.

In a decryption query, which is only available to $\mathcal{A}_1$, the adversary sends a ciphertext, or a superposition of ciphertexts, $\sum_{r,c} |r, c\rangle$ and gets back $\sum_{r,c} |r, c\rangle|c \oplus F(r)\rangle$, from which he can learn $\sum_r |r, F(r)\rangle$. Thus, a decryption query allows $\mathcal{A}_1$ to query F, in superposition.

Later in the challenge phase, $\mathcal{A}_2$ gets an encryption $(r^*, m \oplus F(r^*))$ and must decide if $m = m^*$. Since $\mathcal{A}_2$ no longer has access to the decryption oracle, which allows him to query F, there seem to be two possible ways $\mathcal{A}_2$ could learn $F(r^*)$:

1. $\mathcal{A}_2$ gets lucky in one of his poly(n) queries to Enc_k and happens to sample $(r^*, F(r^*))$;
2. Or, $\mathcal{A}$ is somehow able to use what he learned while he had access to Dec_k, and thus F, to learn $F(r^*)$, meaning that the poly(n)-sized quantum memory $\mathcal{A}_1$ sends to $\mathcal{A}_2$, that can depend on queries to F, but which cannot depend on r^*, allows $\mathcal{A}_2$ to learn $F(r^*)$.

The first possibility is exponentially unlikely, since there are 2^n possibilities for r^*. As we will see shortly, the second possibility would imply a very strong quantum random access code. It would essentially allow $\mathcal{A}_1$ to interact with F, which contains 2^n values, and make a state, which must necessarily be of polynomial size, such that $\mathcal{A}_2$ can use that state to recover $F(r^*)$ for any of the 2^n possible values of r^*, with high probability. We now formalize this intuition. To clarify notation, we will use boldface to denote the shared randomness bitstrings.

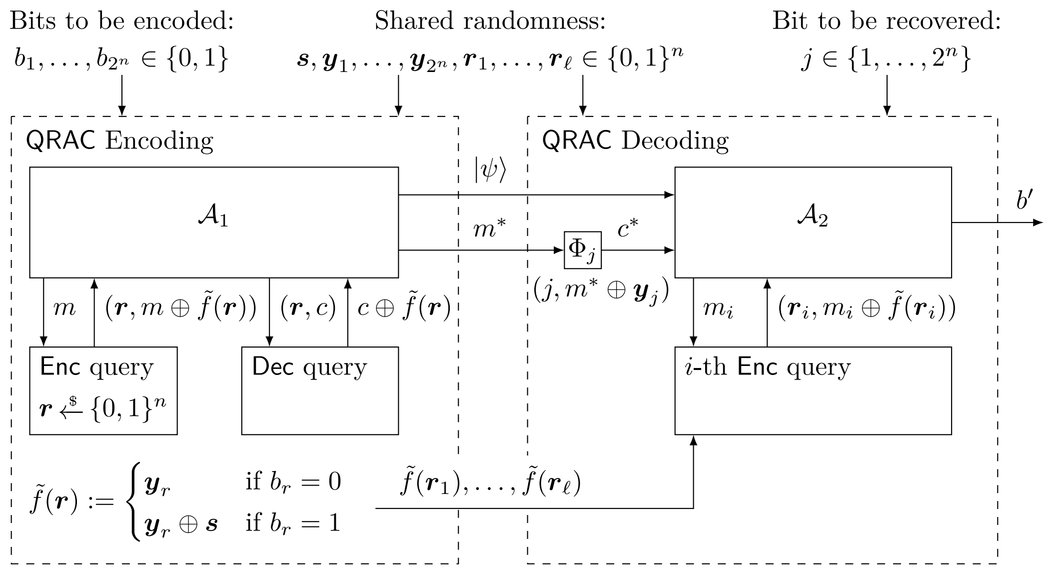

Figure 2 Quantum random access code construction for the PRF scheme.

Construction of a quantum random access code. Let $\mathcal{A}$ be a QPT adversary with winning probability p. Let $\ell = \text{poly}(n)$ be an upper bound on the number of queries made by $\mathcal{A}_2$. Recall that a random access code consists of an encoding procedure that takes (in this case) 2^n bits $b_1, \dots, b_{2^n}$, and outputs a state ϱ of dimension (in this case) $2^{\text{poly}(n)}$, such that a decoding procedure, given ϱ and an index $j \in \{1, \dots, 2^n\}$ outputs b_j with some success probability. We define a quantum random access code as follows (see also Figure 2).

Encoding. Let $b_1, \dots, b_{2^n} \in \{0,1\}$ be the string to be encoded. Let $\boldsymbol{s}, \boldsymbol{y}_1, \dots, \boldsymbol{y}_{2^n} \in \{0,1\}^n$ be the first $n(1+2^n)$ bits of the shared randomness, and let $\boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell \in \{0,1\}^n$ be the next ℓn bits. Define $\tilde{f} : \{0,1\}^n \to \{0,1\}^n$ as follows. For $\boldsymbol{r} \in \{0,1\}^n$, we slightly abuse notation by letting r denote the corresponding integer value between 1 and 2^n. Define $\tilde{f}(\boldsymbol{r}) = \boldsymbol{y}_r \oplus b_r \boldsymbol{s}$. Run $\mathcal{A}_1$, answering encryption and decryption queries using $\tilde{f}$ in place of F. Let m^* and $|\psi\rangle$ be the outputs of $\mathcal{A}_1$ (see Figure 1). Output $\varrho = (|\psi\rangle, m^*, \tilde{f}(\boldsymbol{r}_1), \dots, \tilde{f}(\boldsymbol{r}_\ell))$.

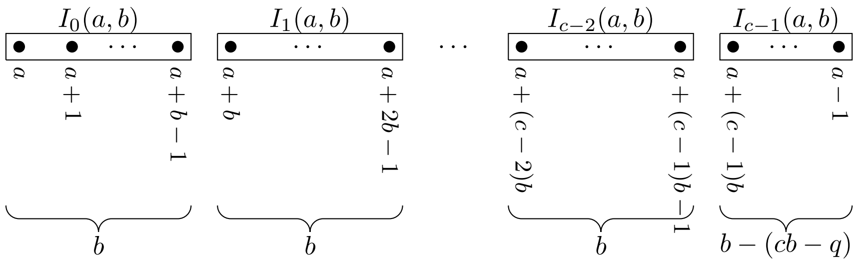

Figure 3 Dividing $\mathbb{Z}_q$ into $c = \lceil q/b \rceil$ blocks, starting from a. The first $c-1$ blocks, labelled $I_0(a,b), \ldots, I_{c-2}(a,b)$, have size b and the last, labelled $I_{c-1}(a,b)$, contains the remaining $b-(cb-q) \leq b$ elements of $\mathbb{Z}_q$.

Decoding. Let $j \in \{1, \ldots, 2^n\}$ be the index of the bit to be decoded (so given ϱ as above, the goal is to recover b_j). Decoding will make use of the values $\boldsymbol{s}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{2^n}, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_\ell$ given by the shared randomness. Upon receiving a query $j \in \{1, \ldots, 2^n\}$, run $\mathcal{A}_2$ with inputs $|\psi\rangle$ and $(j, m^* \oplus \boldsymbol{y}_j)$. On $\mathcal{A}_2$'s i-th encryption oracle call, use randomness $\boldsymbol{r}_i$, so that if the input to the oracle is $|m, c\rangle$, the state returned is $|m, c \oplus (\boldsymbol{r}_i, m \oplus \tilde{f}(\boldsymbol{r}_i))\rangle$ (note that $\tilde{f}(\boldsymbol{r}_i)$ is given as part of ϱ). Return the bit b' output by $\mathcal{A}_2$.

Average bias of the code. We claim that the average probability of decoding correctly, taken over all choices of $b_1, \ldots, b_{2^n} \in \{0,1\}$ and $j \in \{1, \ldots, 2^n\}$, is exactly p, the success probability of $\mathcal{A}$. To see this, first note that from $\mathcal{A}$'s perspective, this is exactly Game 1: the function $\tilde{f}$ is a uniformly random function, and the queries are responded to just as in Game 1. Further, note that if $b_j = 0$, then $m^* \oplus \boldsymbol{y}_j = m^* \oplus \tilde{f}(j)$, so the correct guess for $\mathcal{A}_2$ would be 0, and if $b_j = 1$, then $m^* \oplus \boldsymbol{y}_j = m^* \oplus \tilde{f}(j) \oplus \boldsymbol{s} = \boldsymbol{x} \oplus \tilde{f}(j)$ for the uniformly random string $\boldsymbol{x} = m^* \oplus \boldsymbol{s}$, so the correct guess for $\mathcal{A}_2$ would be 1.

Therefore, the average bias of the code is $p - 1/2$. We also observe that ϱ has dimension at most $2^{\mathrm{poly}(n)}$, since $|\psi\rangle$ must be a $\mathrm{poly}(n)$-qubit state ($\mathcal{A}_1$ only runs for $\mathrm{poly}(n)$ time), and ℓ, the number of queries made by $\mathcal{A}_2$ must be $\mathrm{poly}(n)$, since $\mathcal{A}_2$ only runs for $\mathrm{poly}(n)$ time. As this code encodes 2^n bits into a state of dimension $2^{\mathrm{poly}(n)}$, by Lemma 6 (proven in Appendix A), the bias is $O(2^{-n/2}\,\mathrm{poly}(n)) = \mathrm{negl}(n)$, so $p \leq \frac{1}{2} + \mathrm{negl}(n)$. ◀

5 Quantum algorithm for linear rounding functions

In this section, we analyze the performance of the Bernstein-Vazirani algorithm [3] with a modified version of the oracle. While the original oracle computes the inner product modulo q, our version only gives partial information about it by rounding its value to one of $\lceil q/b \rceil$ blocks of size b, for some $b \in \{1, \ldots, q-1\}$ (if b does not divide q, one of the blocks will have size $< b$).

▶ **Definition 15.** *Let $n \geq 1$ be an integer and $q \geq 2$ be an integer modulus. Let $a \in \mathbb{Z}_q$, $b \in \mathbb{Z}_q \setminus \{0\}$ and $c := \lceil q/b \rceil$. We partition $\mathbb{Z}_q$ into c disjoint blocks (most of them of size b) starting from a as follows (see Figure 3):*

$$I_v(a,b) := \begin{cases} \{a+vb, \ldots, a+vb+b-1\} & \text{if } v \in \{0, \ldots, c-2\}, \\ \{a+vb, \ldots, a+q-1\} & \text{if } v = c-1. \end{cases}$$

Based on this partition, we define a family $\mathsf{LRF}_{\boldsymbol{k},a,b} : \mathbb{Z}_q^n \longrightarrow \mathbb{Z}_c$ *of keyed linear rounding functions, with key* $\boldsymbol{k} \in \mathbb{Z}_q^n$*, as follows:*

$$\mathsf{LRF}_{\boldsymbol{k},a,b}(\boldsymbol{x}) := v \text{ if } \langle \boldsymbol{x}, \boldsymbol{k} \rangle \in I_v(a,b).$$

Algorithm 1: Bernstein-Vazirani for linear rounding functions.

Parameters : n, q, $b \in \{1, \dots, q-1\}$, $c = \lceil q/b \rceil$.
Input : Quantum oracle $U_{\mathsf{LRF}} : |\boldsymbol{x}\rangle|z\rangle \mapsto |\boldsymbol{x}\rangle|z + \mathsf{LRF}_{\boldsymbol{k},a,b}(\boldsymbol{x}) \pmod{c}\rangle$ where $\boldsymbol{x} \in \mathbb{Z}_q^n$, $z \in \mathbb{Z}_c$ and $\mathsf{LRF}_{\boldsymbol{k},a,b}$ is the rounded inner product function for some unknown $\boldsymbol{k} \in \mathbb{Z}_q^n$ and $a \in \mathbb{Z}_q$.
Output : String $\tilde{\boldsymbol{k}} \in \mathbb{Z}_q^n$ such that $\tilde{\boldsymbol{k}} = \boldsymbol{k}$ with high probability.

1. Prepare the uniform superposition and append $\frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z\rangle$ where $\omega_c = e^{2\pi i/c}$:
$$\frac{1}{\sqrt{q^n}} \sum_{\boldsymbol{x} \in \mathbb{Z}_q^n} |\boldsymbol{x}\rangle \otimes \frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z\rangle.$$
2. Query oracle U_{LRF} for $\mathsf{LRF}_{\boldsymbol{k},a,b}$; obtain $\frac{1}{\sqrt{q^n}} \sum_{\boldsymbol{x} \in \mathbb{Z}_q^n} \omega_c^{-\mathsf{LRF}_{\boldsymbol{k},a,b}(\boldsymbol{x})} |\boldsymbol{x}\rangle \otimes \frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z\rangle$.
3. Discard the last register and apply the quantum Fourier transform $\mathsf{QFT}_{\mathbb{Z}_q}^{\otimes n}$.
4. Measure in the computational basis and output the outcome $\tilde{\boldsymbol{k}}$.

The following theorem shows that the modulo-q variant of the Bernstein-Vazirani algorithm (Algorithm 1) can recover $\boldsymbol{k}$ with constant probability of success by using only a single quantum query to $\mathsf{LRF}_{\boldsymbol{k},a,b}$. The proof is a fairly straightforward computation through the steps of the algorithm, and can be found in Appendix C.

▶ **Theorem 16.** *Let* U_{LRF} *be the quantum oracle for the linear rounding function* $\mathsf{LRF}_{\boldsymbol{k},a,b}$ *with modulus* $q \geq 2$*, block size* $b \in \{1, \dots, q-1\}$*, and an unknown* $a \in \{0, \dots, q-1\}$*, and unknown key* $\boldsymbol{k} \in \mathbb{Z}_q^n$ *such that* $\boldsymbol{k}$ *has at least one entry that is a unit modulo* q*. Let* $c = \lceil q/b \rceil$ *and* $d = cb - q$*. By making one query to the oracle* U_{LRF}*, Algorithm 1 recovers the key* $\boldsymbol{k}$ *with probability at least* $4/\pi^2 - O(d/q)$*.*

6 Key recovery against LWE

In this section, we consider various LWE-based encryption schemes and show using Theorem 16 that the decryption key can be efficiently recovered using a single quantum decryption query. In the full version of this work, we also show that a single quantum *encryption* query can be used to recover the secret key in a symmetric-key version of LWE, as long as the querying algorithm also has control over part of the randomness used in the encryption procedure.

6.1 Key recovery via one decryption query in symmetric-key LWE

Recall the following standard construction of an IND-CPA symmetric-key encryption scheme based on the LWE assumption [23].

▶ **Construction 17** (LWE-SKE [23])**.** *Let* $n \geq 1$ *be an integer, let* $q \geq 2$ *be an integer modulus and let* χ *be a discrete and symmetric error distribution. Then, the symmetric-key encryption scheme* $\mathsf{LWE\text{-}SKE}(n, q, \chi) = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is defined as follows:*

1. KeyGen: *output* $\boldsymbol{k} \xleftarrow{\$} \mathbb{Z}_q^n$*;*
2. $\mathsf{Enc}_{\boldsymbol{k}}$: *to encrypt* $b \in \{0,1\}$*, sample* $\boldsymbol{a} \xleftarrow{\$} \mathbb{Z}_q^n$*,* $e \xleftarrow{\chi} \mathbb{Z}_q$ *and output* $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{k} \rangle + b\lfloor \frac{q}{2} \rfloor + e)$*;*
3. $\mathsf{Dec}_{\boldsymbol{k}}$: *to decrypt* $(\boldsymbol{a}, c)$*, output* 0 *if* $|c - \langle \boldsymbol{a}, \boldsymbol{k} \rangle| \leq \lfloor \frac{q}{4} \rfloor$*, else output* 1.

As a corollary of Theorem 16, an adversary that is granted a single quantum decryption query can recover the key with probability at least $4/\pi^2 - o(1)$:

▶ **Corollary 18.** *There is a quantum algorithm that makes a single quantum query to* $\mathsf{LWE\text{-}SKE.Dec}_{\boldsymbol{k}}$ *and recovers the entire key* $\boldsymbol{k}$ *with probability at least* $4/\pi^2 - o(1)$*.*

Proof. $\mathsf{LWE\text{-}SKE.Dec}_{\boldsymbol{k}}$ coincides with a linear rounding function $\mathsf{LRF}_{\boldsymbol{k}',a,b}$ for a key $\boldsymbol{k}' = (-\boldsymbol{k}, 1) \in \mathbb{Z}_q^{n+1}$, which has a unit in its last entry. In particular, $b = \lceil q/2 \rceil$, and if $q = 3 \pmod 4$, $a = \lceil q/4 \rceil$, and otherwise, $a = -\lfloor q/4 \rfloor$. Thus, by Theorem 16, Algorithm 1 makes one quantum query to $\mathsf{LRF}_{\boldsymbol{k}',a,b}$, which can be implemented using one quantum query to $\mathsf{LWE\text{-}SKE.Dec}_{\boldsymbol{k}}$, and recovers $\boldsymbol{k}'$, and thus $\boldsymbol{k}$, with probability $4/\pi^2 - O(d/q)$, where $d = \lceil q/b \rceil b - q \leq 1$. ◀

Note that the key in this scheme consists of $n \log q$ uniformly random bits, and that a classical decryption query yields at most a single bit of output. It follows that any algorithm making t *classical* queries to the decryption oracle recovers the entire key with probability at most $2^{t - n \log q}$. A straightforward key-recovery algorithm does in fact achieve this.

6.2 Key recovery via one decryption query in public-key LWE

The key-recovery attack described in Corollary 18 required nothing more than the fact that the decryption procedure of LWE-SKE is just a linear rounding function whose key contains the decryption key. As a result, the attack is naturally applicable to other variants of LWE. In this section, we consider two public-key variants. The first is the standard construction of IND-CPA public-key encryption based on the LWE assumption, as introduced by Regev [23]. The second is the IND-CPA-secure public-key encryption scheme FrodoPKE [1], which is based on a construction of Lindner and Peikert [18]. In both cases, we demonstrate a dramatic speedup in key recovery using quantum decryption queries.

We emphasize once again that key recovery against these schemes was already possible classically using a linear number of decryption queries. Our results should thus not be interpreted as a weakness of these cryptosystems in their stated security setting (i.e., IND-CPA). The proper interpretation is that, if these cryptosystems are exposed to chosen-ciphertext attacks, then quantum attacks can be even more devastating than classical ones.

6.2.1 Regev's public-key scheme

The standard construction of an IND-CPA public-key encryption scheme based on LWE is the following.

▶ **Construction 19** (LWE-PKE [23])**.** *Let* $m \geq n \geq 1$ *be integers, let* $q \geq 2$ *be an integer modulus, and let* χ *be a discrete error distribution over* $\mathbb{Z}_q$*. Then, the public-key encryption scheme* $\mathsf{LWE\text{-}PKE}(n, q, \chi) = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is defined as follows:*

1. KeyGen: *output a secret key* $\boldsymbol{sk} = \boldsymbol{k} \xleftarrow{\$} \mathbb{Z}_q^n$ *and a public key* $\boldsymbol{pk} = (\boldsymbol{A}, \boldsymbol{Ak} + \boldsymbol{e}) \in \mathbb{Z}_q^{m \times (n+1)}$*, where* $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$*,* $\boldsymbol{e} \xleftarrow{\chi} \mathbb{Z}_q^m$*, and all arithmetic is done modulo* q*.*
2. Enc: *to encrypt* $b \in \{0,1\}$*, pick a random* $\boldsymbol{v} \in \{0,1\}^m$ *with Hamming weight roughly* $m/2$ *and output* $(\boldsymbol{v}^\mathsf{T}\boldsymbol{A}, \boldsymbol{v}^\mathsf{T}(\boldsymbol{Ak} + \boldsymbol{e}) + b\lfloor \frac{q}{2} \rfloor) \in \mathbb{Z}_q^{n+1}$*, where* $\boldsymbol{v}^\mathsf{T}$ *denotes the transpose of* $\boldsymbol{v}$*.*
3. Dec: *to decrypt* $(\boldsymbol{a}, c)$*, output* 0 *if* $|c - \langle \boldsymbol{a}, \boldsymbol{sk} \rangle| \leq \lfloor \frac{q}{4} \rfloor$*, else output* 1.

Although the encryption is now done in a public-key manner, all that matters for our purposes is the decryption procedure, which is identical to the symmetric-key case, LWE-SKE. We thus have the following corollary, whose proof is identical to that of Corollary 18:

▶ **Corollary 20.** *There is a quantum algorithm that makes a single quantum query to* $\mathsf{LWE\text{-}PKE.Dec}_{\boldsymbol{sk}}$ *and recovers the entire key* $\boldsymbol{sk}$ *with probability at least* $4/\pi^2 - o(1)$.

6.2.2 Frodo public-key scheme

Next, we consider the IND-CPA-secure public-key encryption scheme FrodoPKE, which is based on a construction by Lindner and Peikert [18]. Compared to LWE-PKE, this scheme significantly reduces the key-size and achieves better security estimates than the initial proposal by Regev [23]. For a detailed discussion of FrodoPKE, we refer to [1]. We present the entire scheme for completeness, but the important part for our purposes is the decryption procedure.

▶ **Construction 21** (FrodoPKE [1])**.** *Let* $n, \bar{m}, \bar{n}$ *be integer parameters,* $q \geq 2$ *an integer power of 2. Let* B *denote the number of bits used for encoding, and let* χ *be a discrete symmetric error distribution. The public-key encryption scheme* FrodoPKE = (KeyGen, Enc, Dec) *is defined:*

1. KeyGen: *generate a matrix* $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$ *and matrices* $\boldsymbol{S}, \boldsymbol{E} \xleftarrow{\chi} \mathbb{Z}_q^{n\times\bar{n}}$*; compute* $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{E} \in \mathbb{Z}_q^{n\times\bar{n}}$*; output the key-pair* $(\boldsymbol{pk}, \boldsymbol{sk})$ *with public key* $\boldsymbol{pk} = (\boldsymbol{A}, \boldsymbol{B})$ *and secret key* $\boldsymbol{sk} = \boldsymbol{S}$.
2. Enc: *to encrypt* $\boldsymbol{m} \in \{0,1\}^{B\cdot\bar{m}\cdot\bar{n}}$ *(encoded as a matrix* $\boldsymbol{M} \in \mathbb{Z}_q^{\bar{m}\times\bar{n}}$ *with each entry having 0s in all but the* B *most significant bits) with public key* $\boldsymbol{pk}$*, sample error matrices* $\boldsymbol{S}', \boldsymbol{E}' \xleftarrow{\chi} \mathbb{Z}_q^{\bar{m}\times n}$ *and* $\boldsymbol{E}'' \xleftarrow{\chi} \mathbb{Z}_q^{\bar{m}\times\bar{n}}$*; compute* $\boldsymbol{C}_1 = \boldsymbol{S}'\boldsymbol{A} + \boldsymbol{E}' \in \mathbb{Z}_q^{\bar{m}\times n}$ *and* $\boldsymbol{C}_2 = \boldsymbol{M} + \boldsymbol{S}'\boldsymbol{B} + \boldsymbol{E}'' \in \mathbb{Z}_q^{\bar{m}\times\bar{n}}$*; output the ciphertext* $(\boldsymbol{C}_1, \boldsymbol{C}_2)$.
3. Dec: *to decrypt* $(\boldsymbol{C}_1, \boldsymbol{C}_2) \in \mathbb{Z}_q^{\bar{m}\times n} \times \mathbb{Z}_q^{\bar{m}\times\bar{n}}$ *with secret-key* $\boldsymbol{sk} = \boldsymbol{S}$*, compute* $\boldsymbol{M} = \boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{S} \in \mathbb{Z}_q^{\bar{m}\times\bar{n}}$*. For each* $(i,j) \in [\bar{m}] \times [\bar{n}]$*, output the first* B *bits of* $M_{i,j}$.

We now show how to recover $\bar{m}$ of the $\bar{n}$ columns of the secret key $\boldsymbol{S}$ using a single quantum query to $\mathsf{FrodoPKE.Dec}_{\boldsymbol{S}}$. If $\bar{m} = \bar{n}$, as in sample parameters given in [1], then this algorithm recovers $\boldsymbol{S}$ completely.

▶ **Theorem 22.** *There exists a quantum algorithm that makes one quantum query to* $\mathsf{FrodoPKE.Dec}_{\boldsymbol{S}}$ *and recovers any choice of* $\bar{m}$ *of the* $\bar{n}$ *columns of* $\boldsymbol{S}$*. For each of the chosen columns, if that column has at least one odd entry, then the algorithm succeeds in recovering the column with probability at least* $4/\pi^2$.

We give a formal proof of Theorem 22 in Appendix D, but here we briefly sketch the proof. Let $\boldsymbol{s}^1, \dots, \boldsymbol{s}^{\bar{n}}$ be the columns of $\boldsymbol{S}$. Let U denote the map:

$$U : |\boldsymbol{c}\rangle|z_1\rangle \dots |z_{\bar{n}}\rangle \mapsto |\boldsymbol{c}\rangle|z_1 + \mathsf{LRF}_{\boldsymbol{s}^1,0,q/2^B}(\boldsymbol{c})\rangle \dots |z_{\bar{n}} + \mathsf{LRF}_{\boldsymbol{s}^{\bar{n}},0,q/2^B}(\boldsymbol{c})\rangle,$$

for any $\boldsymbol{c} \in \mathbb{Z}_q^n$ and $z_1, \dots, z_{\bar{n}} \in \mathbb{Z}_{2^B}$. By a straightforward calculation, one can show that a single call to $\mathsf{FrodoKEM.Dec}_{\boldsymbol{S}}$, with $\boldsymbol{C}_2$ set of $0^{\bar{m}\times\bar{n}}$, can be used to implement $U^{\otimes\bar{m}}$. Then we show that one call to U can be used to recover any choice of the columns of $\boldsymbol{S}$ with probability $4/\pi^2$, as long as it has at least one entry that is odd. To show this, we show that U can be used to implement a phase query to $\mathsf{LRF}_{\boldsymbol{s}^j,0,q/2^B}$, by simply applying U to a state with $|\varphi\rangle = 2^{-B/2}\sum_{z=0}^{2^B-1}|z\rangle$ in each of $\bar{n}$ registers except the j-th one, and $\frac{1}{\sqrt{2^B}}\sum_{z=0}^{2^B-1}\omega_{2^B}^z|z\rangle$ in the j-th register. This ensures that the phase corresponding to $\mathsf{LRF}_{\boldsymbol{s}^j,0,q/2^B}$ is kicked back, but all other phases, corresponding to $\mathsf{LRF}_{\boldsymbol{s}^{j'},0,q/2^B}$ for $j' \neq j$ are not. For details, see Appendix D.

6.3 Key recovery via one decryption query in public-key Ring-LWE

Next, we analyze key-recovery with a single quantum decryption query against Ring-LWE encryption. Unlike the plain LWE-based encryption schemes we considered in the previous sections, Ring-LWE encryption uses noisy samples over a polynomial ring. In the following, we consider the basic, bit-by-bit Ring-LWE public-key encryption scheme introduced in [19, 20]. It is based on the rings $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and $\mathcal{R}_q := \mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ for some power-of-two integer n and poly(n)-bounded prime modulus q. The details of the error distribution χ below will not be relevant to our results.

▶ **Construction 23** (Ring-LWE-PKE [19, 20]). *Let $n \geq 1$ be an integer, let $q \geq 2$ be an integer modulus, and let χ be an error distribution over $\mathcal{R}$. The public-key encryption scheme* Ring-LWE-PKE = (KeyGen, Enc, Dec) *is defined as follows:*

1. KeyGen*: sample $a \xleftarrow{\$} \mathcal{R}_q$ and $e, s \xleftarrow{\chi} \mathcal{R}$; output $sk = s$ and $pk = (a, c = a\cdot s + e \pmod q) \in \mathcal{R}_q^2$.*
2. Enc*: to encrypt $b \in \{0, 1\}$, sample $r, e_1, e_2 \xleftarrow{\chi} \mathcal{R}$ and output a ciphertext pair $(u, v) \in \mathcal{R}_q^2$, where $u = a \cdot r + e_1 \pmod q$ and $v = c \cdot r + e_2 + b\lfloor q/2\rfloor \pmod q$.*
3. Dec*: to decrypt (u, v), compute $v - u \cdot s = (r \cdot e - s \cdot e_1 + e_2) + b\lfloor q/2\rfloor \pmod q \in \mathcal{R}_q$; output 0 if the constant term of the polynomial is closer to 0 than $\lfloor q/2\rfloor$, else output 1.*

We note that our choice of placing single-bit encryption in the constant term of the polynomial is somewhat arbitrary. Indeed, it is straightforward to extend our results to encryption with respect to other monomials.

In the full version of the article, we show the following corollary to Theorem 16.

▶ **Corollary 24.** *There is a quantum algorithm that makes one quantum query to* Ring-LWE-PKE.Dec$_s$ *and recovers the entire key s with probability at least $4/\pi^2 - o(1)$.*

References

1 Erdem Alkim, Joppe W. Bos, Léo Ducas, Patrick Longa, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Chris Peikert, Ananth Raghunathan, Douglas Stebila, Karen Easterbrook, and Brian LaMacchia. *FrodoKEM – Learning With Errors Key Encapsulation*, 2017. URL: `https://frodokem.org/files/FrodoKEM-specification-20171130.pdf`.

2 Andris Ambainis, Debbie Leung, Laura Mancinska, and Maris Ozols. Quantum random access codes with shared randomness, 2008. `arXiv:0810.2937`.

3 Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. `doi:10.1137/S0097539796300921`.

4 Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO '98*, pages 1–12. Springer, 1998. `doi:10.1007/BFb0055716`.

5 Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, pages 592–608. Springer, 2013. `doi:10.1007/978-3-642-38348-9_35`.

6 Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, pages 361–379. Springer, 2013. `doi:10.1007/978-3-642-40084-1_21`.

7 Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, pages 609–629. Springer, 2015. `doi:10.1007/978-3-662-48000-7_30`.

8 Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Technical report, National Institute of Standards and Technology, 2016. `doi:10.6028/NIST.IR.8105`.

9 Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptography – CRYPTO 1999*, pages 537–554, 1999.

10 Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 60–89. Springer, 2016. doi:10.1007/978-3-662-53015-3_3.

11 Oded Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, Cambridge, UK, 2009.

12 Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. doi:10.1145/6490.6503.

13 Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 25–32, New York, NY, USA, 1989. ACM. doi:10.1145/73007.73010.

14 Alex B. Grilo, Iordanis Kerenidis, and Timo Zijlstra. Learning with errors is easy with quantum samples, 2017. arXiv:1702.08255.

15 Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 207–237. Springer, 2016. doi:10.1007/978-3-662-53008-5_8.

16 Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In *2010 IEEE International Symposium on Information Theory*, pages 2682–2685. IEEE, 2010. doi:10.1109/ISIT.2010.5513654.

17 Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type Even-Mansour cipher. In *2012 International Symposium on Information Theory and its Applications*, pages 312–316. IEEE, 2012. URL: https://ieeexplore.ieee.org/document/6400943/.

18 Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, pages 319–339, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-19074-2_21.

19 Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, pages 35–54. Springer, 2013. doi:10.1007/978-3-642-38348-9_3.

20 Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM*, 60(6):43:1–43:35, November 2013. doi:10.1145/2535925.

21 Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *40th Annual Symposium on Foundations of Computer Science*, pages 369–376, 1999. doi:10.1109/SFFCS.1999.814608.

22 NIST. Post-Quantum Cryptography, 2017. URL: https://csrc.nist.gov/projects/post-quantum-cryptography.

23 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34:1–34:40, 2005. doi:10.1145/1568318.1568324.

24 Thomas Santoli and Christian Schaffner. Using Simon's algorithm to attack symmetric-key cryptographic primitives. *Quantum Information & Computation*, 17(1&2):65–78, 2017. doi:10.26421/QIC17.1-2.

25 Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE, 1994. doi:10.1109/SFCS.1994.365700.

26 Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. doi:10.1137/S0097539796298637.

27 Nicole Tomczak-Jaegermann. The moduli of smoothness and convexity and the Rademacher averages of the trace classes S_p $(1 \leq p < \infty)$. *Studia Mathematica*, 50(2):163–182, 1974. URL: http://eudml.org/doc/217886.

28 Mark Zhandry. How to construct quantum random functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687. IEEE, 2012. doi:10.1109/FOCS.2012.37.
29 Mark Zhandry. A note on quantum-secure PRPs, 2016. arXiv:1611.05564.

A Bound for quantum random access codes

A variation of quantum random access codes allows Alice and Bob to use *shared randomness* in their encoding and decoding operations [2] (note that shared randomness *per se* does not allow them to communicate). We are interested in bounding the average bias $\epsilon = p_{\text{win}} - 1/2$ of a quantum random access code with shared randomness, where p_{win} is the winning probability averaged over $x \xleftarrow{\$} \{0,1\}^N$ and $i \xleftarrow{\$} \{1,\dots,N\}$.

▶ **Lemma 25.** *The average bias of a quantum random access code with shared randomness that encodes N bits into a d-dimensional quantum state is $O(\sqrt{N^{-1}\log d})$. In particular, if $N = 2^n$ and $d = 2^{\text{poly}(n)}$ the bias is $O(2^{-n/2}\,\text{poly}(n))$.*

Proof. A quantum random access code with shared randomness that encodes N bits into a d-dimensional quantum state is specified by the following:

- a shared random variable λ,
- for each $x \in \{0,1\}^N$, a d-dimensional quantum state ϱ_x^λ encoding x,
- for each $i \in \{0,\dots,N\}$, an observable M_i^λ for recovering the i-th bit.

Formally, ϱ_x^λ and M_i^λ are $d \times d$ Hermitian matrices such that $\varrho_x^\lambda \geq 0$, $\text{Tr}\varrho_x^\lambda = 1$, and $\|M_i^\lambda\| \leq 1$ where $\|M_i^\lambda\|$ denotes the operator norm of M_i^λ. Note that both ϱ_x^λ and M_i^λ depend on the shared random variable λ, meaning that Alice and Bob can coordinate their strategies.

The bias of correctly guessing x_i, for a given x and i, is $(-1)^{x_i}\text{Tr}(\varrho_x^\lambda M_i^\lambda)/2$. If the average bias of the code is ϵ then $\mathbb{E}_\lambda \mathbb{E}_{x,i}(-1)^{x_i}\text{Tr}(\varrho_x^\lambda M_i^\lambda) \geq 2\epsilon$. We can rearrange this expression and upper bound each term using its operator norm, and then apply the noncommutative Khintchine inequality [27]:

$$\begin{aligned}\mathbb{E}_\lambda \mathbb{E}_x \frac{1}{N}\text{Tr}\Big(\varrho_x^\lambda \sum_{i=1}^N (-1)^{x_i} M_i^\lambda\Big) &\leq \mathbb{E}_\lambda \mathbb{E}_x \frac{1}{N}\Big\|\sum_{i=1}^N (-1)^{x_i} M_i^\lambda\Big\| \\ &\leq \mathbb{E}_\lambda \frac{1}{N} c\sqrt{N\log d} = c\sqrt{\frac{\log d}{N}},\end{aligned}$$

for some constant c. In other words, $\epsilon \leq \frac{c}{2}\sqrt{\frac{\log d}{N}}$. In the particular case we are interested in, $d = 2^{\text{poly}(n)}$ and $N = 2^n$ so $\epsilon \leq \frac{c}{2}\sqrt{\frac{\text{poly}(n)}{2^n}}$, completing the proof. ◀

B Equivalence of QCCA1 models

Recall that the IND-QCCA1 notion is based on the security game IndGame defined in Definition 9. In the alternative security game IndGame′ (see Definition 10), the adversary provides only one plaintext m and must decide if the challenge is an encryption of m or an encryption of a random string. In this section, we prove the following:

▶ **Proposition 26.** *An encryption scheme Π is* IND-QCCA1 *if and only if for every* QPT $\mathcal{A}$*,*

$$\Pr[\mathcal{A} \text{ wins } \mathsf{IndGame}'(\Pi, \mathcal{A}, n)] \leq 1/2 + \text{negl}(n)\,.$$

Proof. Fix a scheme Π. For one direction, suppose Π is $\mathsf{IND\text{-}QCCA1}$ and let $\mathcal{A}$ be an adversary against $\mathsf{IndGame}'$. Define an adversary $\mathcal{A}_0$ against $\mathsf{IndGame}$ as follows: (i.) run $\mathcal{A}$ until it outputs a challenge plaintext m, (ii.) sample random r and output (m, r), (iii.) run the rest of $\mathcal{A}$ and output what it outputs. The output distribution of $\mathsf{IndGame}'(\Pi, \mathcal{A}, n)$ is then identical to $\mathsf{IndGame}(\Pi, \mathcal{A}_0, n)$, which in turn must be negligibly close to uniform by $\mathsf{IND\text{-}QCCA1}$ security of Π.

For the other direction, suppose no adversary can win $\mathsf{IndGame}'$ with probability better than $1/2$, and let $\mathcal{B}$ be an adversary against $\mathsf{IndGame}$. Now, define two adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ against $\mathsf{IndGame}'$ as follows. The adversary $\mathcal{B}_c$ does: (i.) run $\mathcal{B}$ until it outputs a challenge (m_0, m_1), (ii.) output m_c, (iii.) run the rest of $\mathcal{B}$ and output what it outputs. Note that the pre-challenge algorithm is identical for $\mathcal{B}$, $\mathcal{B}_0$, and $\mathcal{B}_1$; define random variables M_0, M_1 and R given by the two challenges and a uniformly random plaintext, respectively. The post-challenge algorithm is also identical for all three adversaries; call it $\mathcal{C}$. The advantage of $\mathcal{B}$ over random guessing is then bounded by

$$\begin{aligned}
&\|\mathcal{C}(\mathsf{Enc}_k(M_0)) - \mathcal{C}(\mathsf{Enc}_k(M_1))\|_1 \\
&\quad = \|\mathcal{C}(\mathsf{Enc}_k(M_0)) - \mathcal{C}(\mathsf{Enc}_k(M_1)) - \mathcal{C}(\mathsf{Enc}_k(R)) + \mathcal{C}(\mathsf{Enc}_k(R))\|_1 \\
&\quad \leq \|\mathcal{C}(\mathsf{Enc}_k(M_0)) - \mathcal{C}(\mathsf{Enc}_k(R))\|_1 + \|\mathcal{C}(\mathsf{Enc}_k(M_1)) - \mathcal{C}(\mathsf{Enc}_k(R))\|_1 \\
&\quad \leq \mathrm{negl}(n)\,,
\end{aligned}$$

where the last inequality follows from our initial assumption, applied to both $\mathcal{B}_0$ and $\mathcal{B}_1$. It follows that Π is $\mathsf{IND\text{-}QCCA1}$. ◀

C Proof of Theorem 16

In this appendix, we prove Theorem 16, restated below for convenience.

▶ **Theorem 16.** *Let U_{LRF} be the quantum oracle for the linear rounding function $\mathsf{LRF}_{\boldsymbol{k},a,b}$ with modulus $q \geq 2$, block size $b \in \{1, \ldots, q-1\}$, and an unknown $a \in \{0, \ldots, q-1\}$, and unknown key $\boldsymbol{k} \in \mathbb{Z}_q^n$ such that $\boldsymbol{k}$ has at least one entry that is a unit modulo q. Let $c = \lceil q/b \rfloor$ and $d = cb - q$. By making one query to the oracle U_{LRF}, Algorithm 1 recovers the key $\boldsymbol{k}$ with probability at least $4/\pi^2 - O(d/q)$.*

Proof. For an integer m, let $\omega_m = e^{2\pi i/m}$. Several times in this proof, we will make use of the identity $\sum_{z=0}^{\ell-1} \omega_m^{rz} = \omega_m^{r(\ell-1)/2} \left(\frac{\sin(\ell r\pi/m)}{\sin(r\pi/m)}\right)$.

Let $c = \lceil q/b \rfloor$. Throughout this proof, let $\mathsf{LRF}(\boldsymbol{x}) = \mathsf{LRF}_{\boldsymbol{k},a,b}(\boldsymbol{x})$. By querying with $\frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z\rangle$ in the second register, we are using the standard phase kickback technique, which puts the output of the oracle directly into the phase:

$$\begin{aligned}
|\boldsymbol{x}\rangle \frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z\rangle \quad &\overset{U_{\mathsf{LRF}}}{\longmapsto} \quad |\boldsymbol{x}\rangle \frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z + \mathsf{LRF}(\boldsymbol{x}) \pmod{c}\rangle \\
&= \quad |\boldsymbol{x}\rangle \frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^{z-\mathsf{LRF}(\boldsymbol{x})} |z\rangle = \omega_c^{-\mathsf{LRF}(\boldsymbol{x})} |\boldsymbol{x}\rangle \frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z\rangle.
\end{aligned}$$

Thus, after querying the uniform superposition over the cipherspace with $\frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z\rangle$ in the second register, we arrive at the state

$$\frac{1}{\sqrt{q^n}} \sum_{\boldsymbol{x} \in \mathbb{Z}_q^n} \omega_c^{-\mathsf{LRF}(\boldsymbol{x})} |\boldsymbol{x}\rangle \frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z\rangle.$$

Note that $\omega_c = \omega_q^{q/c}$. If we discard the last register and apply $\mathsf{QFT}_{\mathbb{Z}_q}^{\otimes n}$, we get

$$|\psi\rangle = \frac{1}{q^n} \sum_{\boldsymbol{y}\in\mathbb{Z}_q^n} \sum_{\boldsymbol{x}\in\mathbb{Z}_q^n} \omega_q^{-(q/c)\mathsf{LRF}(\boldsymbol{x})+\langle \boldsymbol{x},\boldsymbol{y}\rangle} |\boldsymbol{y}\rangle.$$

We then perform a complete measurement in the computational basis. The probability of obtaining the key $\boldsymbol{k}$ is given by

$$|\langle \boldsymbol{k}|\psi\rangle|^2 = \left| \frac{1}{q^n} \sum_{\boldsymbol{x}\in\mathbb{Z}_q^n} \omega_q^{-\frac{q}{c}\mathsf{LRF}(\boldsymbol{x})+\langle \boldsymbol{x},\boldsymbol{k}\rangle} \right|^2 = \left| \frac{1}{q^n} \sum_{v=0}^{c-1} \omega_q^{-\frac{q}{c}v} \sum_{\boldsymbol{x}\in\mathbb{Z}_q^n:\mathsf{LRF}(\boldsymbol{x})=v} \omega_q^{\langle \boldsymbol{x},\boldsymbol{k}\rangle} \right|^2. \tag{2}$$

We are assuming that $\boldsymbol{k}$ has at least one entry that is a unit modulo q. For simplicity, suppose that entry is k_n. Let $\boldsymbol{k}_{1:n-1}$ denote the first $n-1$ entries of $\boldsymbol{k}$. Then, for any $v \in \{0,\dots,c-2\}$:

$$\begin{aligned} \sum_{\boldsymbol{x}\in\mathbb{Z}_q^n:\mathsf{LRF}(\boldsymbol{x})=v} \omega_q^{\langle \boldsymbol{x},\boldsymbol{k}\rangle} &= \sum_{\boldsymbol{x}\in\mathbb{Z}_q^n:\langle \boldsymbol{x},\boldsymbol{k}\rangle\in I_v(a,b)} \omega_q^{\langle \boldsymbol{x},\boldsymbol{k}\rangle} \\ &= \sum_{\boldsymbol{y}\in\mathbb{Z}_q^{n-1}} \omega_q^{\langle \boldsymbol{y},\boldsymbol{k}_{1:n-1}\rangle} \sum_{\substack{x_n\in\mathbb{Z}_q: \\ x_n k_n\in I_v(a-\langle \boldsymbol{y},\boldsymbol{k}_{1:n-1}\rangle,b)}} \omega_q^{x_n k_n}. \end{aligned} \tag{3}$$

(Recall the definition of $I_v(a,b)$ from Definition 15). Since k_n is a unit, for each $z \in I_v(a-\langle \boldsymbol{y},\boldsymbol{k}_{1:n-1}\rangle)$, there is a unique $x_n \in \mathbb{Z}_q$ such that $x_n k_n = z$. Thus, for a fixed $\boldsymbol{y} \in \mathbb{Z}_q^{n-1}$, letting $a' = a - \langle \boldsymbol{y},\boldsymbol{k}_{1:n-1}\rangle$, we have:

$$\sum_{x_n\in\mathbb{Z}_q:x_n k_n\in I_v(a',b)} \omega_q^{x_n k_n} = \sum_{z=a'+vb}^{a'+(v+1)b-1} \omega_q^z = \omega_q^{a'+vb} \sum_{z=0}^{b-1} \omega_q^z,$$

which we can plug into (3) to get:

$$\sum_{\substack{\boldsymbol{x}\in\mathbb{Z}_q^n: \\ \mathsf{LRF}(\boldsymbol{x})=v}} \omega_q^{\langle \boldsymbol{x},\boldsymbol{k}\rangle} = \sum_{\boldsymbol{y}\in\mathbb{Z}_q^{n-1}} \omega_q^{\langle \boldsymbol{y},\boldsymbol{k}_{1:n-1}\rangle} \omega_q^{a-\langle \boldsymbol{y},\boldsymbol{k}_{1:n-1}\rangle+vb} \sum_{z=0}^{b-1} \omega_q^z = q^{n-1}\omega_q^{a+vb} \sum_{z=0}^{b-1} \omega_q^z. \tag{4}$$

We can perform a similar analysis for the remaining case when $v = c-1$. Recall that $d = cb - q \geq 0$ so $vb = cb - b = d + q - b = -(b-d) \pmod q$ and we get

$$\sum_{\boldsymbol{x}\in\mathbb{Z}_q^n:\mathsf{LRF}(\boldsymbol{x})=c-1} \omega_q^{\langle \boldsymbol{x},\boldsymbol{k}\rangle} = q^{n-1}\omega_q^{a-(b-d)} \sum_{z=0}^{b-d-1} \omega_q^z. \tag{5}$$

This is slightly different from the $v < c-1$ case, shown in (4), but very similar. If we substitute $v = c-1$ in (4) and compare it to (5), we get

$$\begin{aligned} & \left| q^{n-1}\omega_q^{a-(b-d)} \sum_{z=0}^{b-d-1} \omega_q^z - q^{n-1}\omega_q^{a-(b-d)} \sum_{z=0}^{b-1} \omega_q^z \right| \\ = \quad & q^{n-1} \left| \sum_{z=b-d}^{b-1} \omega_q^z \right| = q^{n-1} \left| \sum_{z=0}^{d-1} \omega_q^z \right| = q^{n-1} \left| \frac{\sin(\pi d/q)}{\sin(\pi/q)} \right| \\ \leq \quad & q^{n-1} \frac{\pi d/q}{2/q} = q^{n-1}\frac{\pi}{2} d. \end{aligned} \tag{6}$$

Above, we have used the facts $\sin x \leq x$, and $|\sin x| \geq 2x/\pi$ when $|x| \leq \pi/2$. Now, plugging (4) into (2) for all the $v < c-1$ terms, and using (6) and the triangle inequality for the $v = c-1$ term, we get:

$$\begin{aligned} |\langle \boldsymbol{k}|\psi\rangle| &\geq \left|\frac{1}{q^n}\sum_{v=0}^{c-1}\omega_q^{-qv/c}\cdot q^{n-1}\omega_q^{a+vb}\sum_{z=0}^{b-1}\omega_q^z\right| - \left|\frac{1}{q^n}\omega_q^{-q(c-1)/c}\cdot q^{n-1}\frac{\pi}{2}d\right| \\ &= \frac{1}{q}\left|\sum_{v=0}^{c-1}\omega_q^{v(b-q/c)}\frac{\sin(b\pi/q)}{\sin(\pi/q)}\right| - \frac{\pi}{2}\frac{d}{q} \\ &= \frac{1}{q}\frac{\sin(b\pi/q)}{\sin(\pi/q)}\left|\sum_{v=0}^{c-1}\omega_q^{v(b-q/c)}\right| - \frac{\pi}{2}\frac{d}{q}. \end{aligned} \tag{7}$$

Since $b - q/c = d/c$, we can bound the sum as follows:

$$\begin{aligned} \left|\sum_{v=0}^{c-1}\omega_q^{v(b-q/c)}\right| = \left|\sum_{v=0}^{c-1}\omega_q^{vd/c}\right| &\geq \left|\sum_{v=0}^{c-1}\cos\left(\frac{2\pi}{q}\frac{vd}{c}\right)\right| \\ &\geq \left|\sum_{v=0}^{c-1}\cos\left(\frac{2\pi}{q}d\right)\right| = \left|c\cos\left(\frac{2\pi d}{q}\right)\right| \qquad (8) \\ &\geq c\sqrt{1-(2\pi d/q)^2}. \qquad (9) \end{aligned}$$

To get the inequality (8), we used $0 \leq v \leq c$ and the assumption that $d/q \leq 1/4$ (if $d/q > 1/4$, the claim of the theorem is trivial), which implies that $\frac{2\pi v}{c}\frac{d}{q} \leq \frac{\pi}{2}$. The last inequality follows from $|\cos x| \geq \sqrt{1-x^2}$.

Next, we bound $\frac{\sin(b\pi/q)}{\sin(\pi/q)}$. When $b/q \leq 1/2$, $b\pi/q \leq \pi/2$, so we have $\sin(b\pi/q) \geq 2b/q$. We also have $\sin(\pi/q) \leq \pi/q$. Thus,

$$\frac{\sin(b\pi/q)}{\sin(\pi/q)} \geq \frac{2b}{\pi}.$$

On the other hand, when $b/q > 1/2$, we must have $c = 2$ and $b = \frac{q+d}{2}$. In that case

$$\sin(b\pi/q) = \sin\frac{\pi(q+d)}{2q} = \sin\left(\frac{\pi}{2}+\frac{\pi}{2}\frac{d}{q}\right) = \cos\frac{\pi d}{2q} \geq \sqrt{1-\left(\frac{\pi d}{2q}\right)^2}.$$

Since $\sin(\pi/q) \leq \pi/q$ and $q \geq 2b$,

$$\frac{\sin(b\pi/q)}{\sin(\pi/q)} \geq \frac{\sqrt{1-\left(\frac{\pi d}{2q}\right)^2}}{\pi/q} \geq \frac{2b}{\pi}\sqrt{1-O(d/q)}.$$

Thus, in both cases, $\frac{\sin(b\pi/q)}{\sin(\pi/q)} \geq \frac{2b}{\pi}\sqrt{1-O(d/q)}$. Plugging this and (9) into (7), we get:

$$\begin{aligned} |\langle \boldsymbol{k}, \psi\rangle| &\geq \frac{1}{q}\cdot\frac{2b}{\pi}\sqrt{1-O(d/q)}\cdot c\sqrt{1-O(d/q)} - O(d/q) \\ &= \frac{2}{\pi}\frac{bc}{q} - O(d/q) = \frac{2}{\pi}\frac{q+d}{q} - O(d/q) = \frac{2}{\pi} - O(d/q), \end{aligned}$$

completing the proof. ◀

D Proof of Theorem 22

In this appendix, we prove Theorem 22, restated below for convenience.

▶ **Theorem 22.** *There exists a quantum algorithm that makes one quantum query to* $\mathsf{FrodoPKE.Dec}_{\boldsymbol{S}}$ *and recovers any choice of* $\bar{m}$ *of the* $\bar{n}$ *columns of* $\boldsymbol{S}$*. For each of the chosen columns, if that column has at least one odd entry, then the algorithm succeeds in recovering the column with probability at least* $4/\pi^2$*.*

Proof. Let $\boldsymbol{s}^1, \ldots, \boldsymbol{s}^{\bar{n}}$ be the columns of $\boldsymbol{S}$. Let U denote the map:

$$U : |\boldsymbol{c}\rangle|z_1\rangle \ldots |z_{\bar{n}}\rangle \mapsto |\boldsymbol{c}\rangle|z_1 + \mathsf{LRF}_{\boldsymbol{s}^1,0,q/2^B}(\boldsymbol{c})\rangle \ldots |z_{\bar{n}} + \mathsf{LRF}_{\boldsymbol{s}^{\bar{n}},0,q/2^B}(\boldsymbol{c})\rangle,$$

for any $\boldsymbol{c} \in \mathbb{Z}_q^n$ and $z_1, \ldots, z_{\bar{n}} \in \mathbb{Z}_{2^B}$. We first argue that one call to $\mathsf{FrodoKEM.Dec}_{\boldsymbol{S}}$ can be used to implement $U^{\otimes \bar{m}}$. Then we show that one call to U can be used to recover any choice of the columns of $\boldsymbol{S}$ with probability $4/\pi^2$, as long as it has at least one entry that is odd.

Let $\mathsf{Trunc} : \mathbb{Z}_q \mapsto \mathbb{Z}_{2^B}$ denote the map that takes $x \in \mathbb{Z}_q$ to the integer represented by the B most significant bits of the binary representation of x. We have, for any $\boldsymbol{C}_1 \in \mathbb{Z}_q^{\bar{m}\times n}$, $\boldsymbol{C}_2 = 0^{\bar{m}\times\bar{n}}$, and any $\{z_{i,j}\}_{i\in[\bar{m}],j\in[\bar{n}]} \subseteq \mathbb{Z}_{2^B}$:

$$U_{\mathsf{FrodoKEM.Dec}} : |\boldsymbol{C}_1\rangle|0^{\bar{m}\cdot\bar{n}}\rangle \bigotimes_{i\in[\bar{m}],j\in[\bar{n}]} |z_{i,j}\rangle \mapsto |\boldsymbol{C}_1\rangle|0^{\bar{m}\cdot\bar{n}}\rangle \bigotimes_{i\in[\bar{m}],j\in[\bar{n}]} |z_{i,j} + \mathsf{Trunc}([\boldsymbol{C}_1\boldsymbol{S}]_{i,j})\rangle. \tag{10}$$

Above, $[\boldsymbol{C}_1\boldsymbol{S}]_{i,j}$ represents the ij-th entry of $\boldsymbol{C}_1\boldsymbol{S}$. If $\boldsymbol{c}^1, \ldots, \boldsymbol{c}^{\bar{m}}$ denote the rows of $\boldsymbol{C}_1$, then $[\boldsymbol{C}_1\boldsymbol{S}]_{i,j} = \langle \boldsymbol{c}^i, \boldsymbol{s}^j\rangle$. Thus, $\mathsf{Trunc}([\boldsymbol{C}_1\boldsymbol{S}]_{i,j}) = \mathsf{LRF}_{\boldsymbol{s}^j,0,q/2^B}(\boldsymbol{c}^i)$, the linear rounding function with block size $b = q/2^B$, which is an integer since q is a power of 2, and $a = 0$. Note that we have also assumed that the plaintext is *subtracted* rather than added to the last register; this is purely for convenience of analysis, and can easily be accounted for by adjusting Algorithm 1 (e.g., by using inverse-QFT instead of QFT.)

Discarding the second register (containing $\boldsymbol{C}_2 = 0$), the right-hand side of (10) becomes

$$|\boldsymbol{c}^1\rangle \ldots |\boldsymbol{c}^{\bar{m}}\rangle \bigotimes_{i\in[\bar{m}],j\in[\bar{n}]} |z_{i,j} + \mathsf{LRF}_{\boldsymbol{s}^j,0,q/2^B}(\boldsymbol{c}^i)\rangle. \tag{11}$$

Reordering the registers of (11), we get:

$$\bigotimes_{i\in[\bar{m}]} \left(|\boldsymbol{c}^i\rangle \bigotimes_{j\in[\bar{n}]} |z_{i,j} + \mathsf{LRF}_{\boldsymbol{s}^j,0,q/2^B}(\boldsymbol{c}^i)\rangle \right) = U^{\otimes\bar{m}} \left(\bigotimes_{i\in[\bar{m}]} |\boldsymbol{c}^i\rangle \bigotimes_{j\in[\bar{n}]} |z_{i,j}\rangle \right).$$

Thus, we can implement $U^{\otimes\bar{m}}$ using a single call to $\mathsf{FrodoKEM.Dec}_{\boldsymbol{S}}$.

Next we show that for any particular $j \in [\bar{n}]$, a single call to U can be used to recover $\boldsymbol{s}^j$, the j-th column of $\boldsymbol{S}$, with probability at least $4/\pi^2$, as long as at least one entry of $\boldsymbol{s}^j$ is odd. To do this, we show how one use of U can be used to implement one phase query to $\mathsf{LRF}_{s^j,0,q/2^B}$. Then the result follows from the proof of Theorem 16.

Let $|\varphi\rangle = 2^{-B/2}\sum_{z=0}^{2^B-1}|z\rangle$, and define

$$|\phi_j\rangle = |\varphi\rangle^{\otimes(j-1)} \otimes \frac{1}{\sqrt{2^B}} \sum_{z=0}^{2^B-1} \omega_{2^B}^z |z\rangle \otimes |\varphi\rangle^{\otimes(\bar{n}-j)}.$$

Then for any $\boldsymbol{c} \in \mathbb{Z}_q^n$, we have:

$$\frac{1}{\sqrt{2^B}} \sum_{z=0}^{2^B-1} |z + \mathsf{LRF}_{\boldsymbol{s}^i,0,q/2^B}(\boldsymbol{c})\rangle = \frac{1}{\sqrt{2^B}} \sum_{z=0}^{2^B-1} |z\rangle = |\varphi\rangle,$$

since addition here is modulo 2^B, and

$$\frac{1}{\sqrt{2^B}}\sum_{z=0}^{2^B-1}\omega_{2^B}^{z}|z+\mathsf{LRF}_{\boldsymbol{s}^j,0,q/2^B}(\boldsymbol{c})\rangle=\frac{1}{\sqrt{2^B}}\sum_{z=0}^{2^B-1}\omega_{2^B}^{z-\mathsf{LRF}_{\boldsymbol{s}^j,0,q/2^B}(\boldsymbol{c})}|z\rangle.$$

Thus:

$$\begin{aligned}U(|\boldsymbol{c}\rangle|\phi_j\rangle) &= |\boldsymbol{c}\rangle|\varphi\rangle^{\otimes(j-1)}\otimes\frac{1}{\sqrt{2^B}}\sum_{z=0}^{2^B-1}\omega_{2^B}^{z-\mathsf{LRF}_{\boldsymbol{s}^j,0,q/2^B}(\boldsymbol{c})}|z\rangle\otimes|\varphi\rangle^{\otimes(\bar{n}-j)}\\ &= \omega_{2^B}^{-\mathsf{LRF}_{\boldsymbol{s}^j,0,q/2^B}(\boldsymbol{c})}|\boldsymbol{c}\rangle|\phi_j\rangle.\end{aligned}$$

Thus, by the proof of Theorem 16, if we apply U to $q^{-n/2}\sum_{\boldsymbol{c}\in\mathbb{Z}_q^n}|\boldsymbol{c}\rangle|\phi_j\rangle$, Fourier transform the first register, and then measure, assuming $\boldsymbol{s}^j$ has at least one entry that is a unit[3] we will measure $\boldsymbol{s}^j$ with probability at least $\pi^2/4-O(d/q)$, where $d=q/2^B\lceil q/(q/2^B)\rceil-q=0$.

Thus, if we want to recover columns $j_1,\dots j_{\bar{m}}$ of $\boldsymbol{S}$, we apply our procedure for $U^{\otimes\bar{m}}$, which costs one query to $\mathsf{FrodoKEM.Dec}_{\boldsymbol{S}}$, to the state

$$\sum_{\boldsymbol{c}\in\mathbb{Z}_q^n}\frac{1}{\sqrt{q^n}}|\boldsymbol{c}\rangle|\phi_{j_1}\rangle\otimes\cdots\otimes\sum_{\boldsymbol{c}\in\mathbb{Z}_q^n}\frac{1}{\sqrt{q^n}}|\boldsymbol{c}\rangle|\phi_{j_{\bar{m}}}\rangle,$$

Fourier transform each of the $\boldsymbol{c}$ registers, and then measure. ◀

[3] since q is a power of 2, this is just an odd number