



Article

On Quantum Chosen-Ciphertext Attacks and Learning with Errors [†]

Gorjan Alagic ^{1,2,*}, Stacey Jeffery ^{3,4}, Maris Ozols ^{4,5} and Alexander Poremba ^{6,*}

¹ Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD 20742, USA

² National Institute of Standards and Technology, Gaithersburg, MD 20899, USA

³ Centrum Wiskunde & Informatica, 1098 XG Amsterdam, The Netherlands; smjeffery@gmail.com

⁴ QuSoft, 1098 XG Amsterdam, The Netherlands

⁵ ILLC, KdVI, and IoP, University of Amsterdam, 1012 WN Amsterdam, The Netherlands; marozols@gmail.com

⁶ Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA

* Correspondence: galagic@gmail.com (G.A.); aporemba@caltech.edu (A.P.)

[†] This paper is an extended version of our paper published in the Proceedings of the 14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC) 2019, College Park, MD, USA, 3–5 June 2019.

Received: 16 December 2019; Accepted: 18 March 2020; Published: 21 March 2020



Abstract: Large-scale quantum computing poses a major threat to classical public-key cryptography. Recently, strong “quantum access” security models have shown that numerous symmetric-key cryptosystems are also vulnerable. In this paper, we consider classical encryption in a model that grants the adversary quantum oracle access to encryption and decryption, but where we restrict the latter to non-adaptive (i.e., pre-challenge) queries only. We formalize this model 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. We show that the standard pseudorandom function (PRF)-based encryption schemes are QCCA1-secure when instantiated with quantum-secure primitives. Our security proofs use a strong bound on quantum random-access codes with shared randomness. Revisiting plain IND-CPA-secure Learning with Errors (LWE) encryption, we show that leaking only a single quantum decryption query (and no other leakage or queries of any kind) allows the adversary to recover the full secret key with constant success probability. Information-theoretically, full recovery of the key in the classical setting requires at least a linear number of decryption queries. Our results thus challenge the notion that LWE is unconditionally “just as secure” quantumly as it is classically. The algorithm at the core of our attack is a new variant of the well-known Bernstein–Vazirani algorithm. Finally, 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.

Keywords: quantum chosen-ciphertext security; quantum attacks; key-recovery; learning with errors

1. Introduction

1.1. Background

Full-fledged quantum computers with millions of logical qubits threaten classical cryptography dramatically. 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 [1,2]. 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 [3] and its variants [1,4]. The LWE problem is closely related to lattice problems and is widely believed to be intractable even for quantum computers. It thus forms the basis for a number of candidate post-quantum cryptosystems. In addition to motivating significant work on post-quantum cryptosystems, the threat of quantum computers has also spurred general research on secure classical cryptography in the presence of quantum adversaries. One area in particular explores strong security models where a quantum adversary gains precise 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 [5–9]. It is unclear if the assumption behind these models is plausible for typical physical implementations of symmetric-key cryptography. However, attacks which 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 certainly the case for hashing, public-key encryption, and circuit obfuscation. Moreover, understanding this model is crucial for gauging the degree to which any physical device involved in cryptography must be resistant to reverse engineering or forced quantum behavior (consider, e.g., 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. One devastating attack that allows an adversary to exercise precise control over a cryptosystem is characterized by a chosen-ciphertext attack (CCA). In this scenario, the attacker receives *adaptive* access to the encryption and decryption procedures of a cryptosystem, i.e., the attacker is able to choose plaintexts and ciphertexts that depend on previous outcomes of the attack. Despite their strong security requirements, chosen-ciphertext attacks are often perfectly realistic. Surprisingly, even widely adopted cryptosystems such as RSA have been shown to be vulnerable. For example, a well-known attack due to Bleichenbacher [11] only requires access to an oracle that decides if the input ciphertext is encrypted according to a particular RSA standard.

In this work, we 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 [12,13], encryption secure against quantum chosen-plaintext attacks (QCPA) [10,14], and encryption secure against *adaptive* quantum chosen-ciphertext attacks (QCCA2) [12]. Gagliardoni, Hülsing and Schaffner also consider a wide range of indistinguishability and semantic-security models within the QCPA framework [10].

1.2. Our Contributions

1.2.1. The Model

In this work, we consider 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 theoretical cryptography. In fact, CPA and CCA2 are intermediate models themselves, since they are both strictly weaker than authenticated encryption. Our particular intermediate model is naturally motivated: it is sufficient 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, where the adversary supplies two challenge plaintexts (m_0, m_1) , receives a challenge ciphertext $\text{Enc}_k(m_b)$ for random b , and must correctly guess b . Alternatively, the challenge can be semantic, where the adversary receives partial information about a plaintext m , and is tasked with outputting some *additional* information about m by making use of its encryption $\text{Enc}_k(m)$. 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 define both IND-QCCA1 and SEM-QCCA1 formally, and prove that they are equivalent [10,12,14].

We then prove IND-QCCA1 security for two symmetric-key encryption schemes, based on standard assumptions. Specifically, we show that the standard encryption schemes based on quantum-secure pseudorandom functions (QPRF) and quantum-secure pseudorandom permutations (QPRP) are both IND-QCCA1. We remark that both QPRFs and QPRPs can be constructed from quantum-secure one-way functions [15,16]. 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.

1.2.2. 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 [17]. 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 [3], the FrodoPKE scheme (FrodoPKE is an IND-CPA-secure building block in the IND-CCA2-secure post-quantum cryptosystem “FrodoKEM” [18]. Our results do not affect the post-quantum security of Frodo and do not contradict the CCA2 security of FrodoKEM.) [18,19], and standard Ring-LWE [20,21].

Finally, we consider a setting where the adversary receives one quantum encryption query, including access to the randomness register. We show that a similar algorithm again leads to complete key recovery in this model. In this case, the analysis amounts to the observation of Grilo et al. that the LWE problem becomes easy with access to quantum samples [22].

The novelty of our results on LWE is as follows. First, we show that a variant of the Bernstein–Vazirani algorithm succeeds even if the parameters correspond to a broad cryptographic security model; second, we show that this leads to a dramatic quantum speed-up in an actual attack: full key-recovery with a single quantum query in each case.

1.2.3. Important Caveats

Our results challenge the view that LWE encryption is “just as secure” classically as it is quantum in any real-world application; notably, this intuition stems from the conjecture that quantum computers do not seem to solve the standard LWE problem much faster than classical computers. 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 plain LWE-based encryption schemes we consider in this work are already vulnerable to decryption queries in a purely classical attack model. In particular, we remark that one classical decryption query can produce at most one bit of the $(n \log q)$ -bit key. Similarly, a classical encryption query which can access also the randomness used in the encryption can produce at most $\log q$ bits of the key. 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. Finally, we remark that schemes we attack can be modified to achieve chosen-ciphertext security [23].

1.2.4. 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 [22]. 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 [22] 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.

1.3. Technical Summary of Results

1.3.1. Security Model and Basic Definitions

First, we set down the basic QCCA1 security model, adapting the ideas of [10,13]. Recall that an encryption scheme is a triple $\Pi = (\text{KeyGen}, \text{Enc}, \text{Dec})$ of algorithms (key generation, encryption, and decryption, respectively) satisfying $\text{Dec}_k(\text{Enc}_k(m)) = m$ for any key $k \leftarrow \text{KeyGen}$ and message m . Here, KeyGen represents a procedure for generating a key, Enc represents the encryption procedure, and Dec the decryption procedure. In what follows, all oracles are quantum, meaning that a function f is accessed via the unitary operator $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$. We define ciphertext 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 $1/2 + \text{negl}(n)$.

1. a key $k \leftarrow \text{KeyGen}(1^n)$ 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} receives $\text{Enc}_k(m_b)$ and gets access to an oracle for Enc_k only, 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 $k \leftarrow \text{KeyGen}(1^n)$ is generated; \mathcal{A} gets access to oracles $\text{Enc}_k, \text{Dec}_k$ and outputs circuits (Samp, h, f) ;
2. Sample $m \leftarrow \text{Samp}$; \mathcal{A} receives $h(m), \text{Enc}_k(m)$, 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 $\text{Enc}_k(m)$ in step 2.

The following fact is straightforward.

Theorem 1. *A classical symmetric-key encryption scheme is IND-QCCA1 if and only if it is SEM-QCCA1.*

1.3.2. 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 [15]. Given a PRF $f = \{f_k\}_k$, define $\text{PRFscheme}[f]$ to be the scheme which encrypts a plaintext m using randomness r via $\text{Enc}_k(m; r) = (r, f_k(r) \oplus m)$ and decrypts in the obvious way.

Theorem 2. *If f is a QPRF, then $\text{PRFscheme}[f]$ is IND-QCCA1-secure.*

We also analyze a standard permutation-based scheme. Quantum-secure PRPs (i.e., QPRPs) can be obtained from quantum-secure one-way functions [16]. Given a PRP $P = \{P_k\}_k$, define $\text{PRPscheme}[P]$ to be the scheme that encrypts a plaintext m using randomness r via $\text{Enc}_k(m; r) = P_k(m||r)$, where $||$ denotes concatenation; to decrypt, one applies P_k^{-1} and discards the randomness bits. We show the following.

Theorem 3. *If P is a QPRP, then $\text{PRPscheme}[P]$ is IND-QCCA1-secure.*

We briefly describe our proof techniques for Theorems 2 and 3. 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 their private, poly-sized quantum memory. From this point of view, establishing security means showing 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). Informally, a QRAC is a mapping from N -bit strings x to d -dimensional quantum states q_x , such that given q_x , and any index $j \in [N]$, the bit x_j can be recovered with some probability $p_{x,j} = \frac{1}{2} + \epsilon_{x,j}$. The average bias of such a code is the expected value of $\epsilon_{x,j}$, over uniform x and j . A QRAC with shared randomness further allows the encoding and decoding procedures to both depend on some random variable.

Lemma 1. *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 $O(2^{-n/2} \text{poly}(n))$.*

The proof of Theorem 3 is similar to that of Theorem 2.

1.3.3. Key Recovery Against LWE

Our attack on LWE encryption will make use of a new analysis of the performance of a large-modulus variant of the Bernstein–Vazirani algorithm [17], in the presence of a certain type of “rounding” noise.

1.3.4. Quantum Algorithm for Linear Rounding Functions

In the simplest case we analyze, the oracle outputs 0 if the inner product is small, and 1 otherwise. Specifically, given integers $n \geq 1$ and $q \geq 2$, we consider keyed families of (binary) linear rounding functions, $\text{LRF}_{k,q} : \mathbb{Z}_q^n \rightarrow \{0, 1\}$, with key $k \in \mathbb{Z}_q^n$, as follows:

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

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

Theorem 4 (informal). *There exists a quantum algorithm which runs in time $O(n)$, makes one quantum query to $\text{LRF}_{k,q}$ (with $q \geq 2$ and unknown $k \in \mathbb{Z}_q^n$), and outputs k with probability at least $(\frac{2}{\pi} - \frac{\pi}{8})^2 \approx 0.059$.*

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.

1.3.5. 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 $(a, c) \in \mathbb{Z}_q^{n+1}$ with key k by outputting 0 if $|c - \langle a, k \rangle| \leq \lfloor \frac{q}{4} \rfloor$ 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 is 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 4.

Theorem 5 (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 Dec_k of Π , and outputs the secret key with constant probability.*

1.4. 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 and PRP 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 some rounded value of $\langle k, x \rangle$ (i.e., the oracle is a linear rounding function). 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; we also observe a similar result for a model in which the adversary can make one quantum encryption query with partial access to the randomness register.

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., $v \xleftarrow{\chi} \mathbb{Z}_q^n$. Given a matrix A , A^T will denote the transpose of A . We will view elements v of \mathbb{Z}_q^n as column vectors; The notation v^T then denotes the corresponding row vector. We use bold fonts to discriminate between vectors (or matrices) and scalars. The Kronecker delta δ_{ij} of two nonnegative integers i and j is equal to 1 if $i = j$, and 0 otherwise. The notation $\text{negl}(n)$ 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. The notation $\Pr[1 \leftarrow \text{GAME } 0]$ denotes the probability that a QPT \mathcal{A} outputs 1 in some security game GAME 0.

Every function $f : \{0, 1\}^m \rightarrow \{0, 1\}^\ell$ determines a unitary operator $U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$ on $m + \ell$ qubits where $x \in \{0, 1\}^m$ and $y \in \{0, 1\}^\ell$. In this work, when we say that a quantum algorithm \mathcal{A} gets (adaptive) oracle access to f (written \mathcal{A}^f), we mean that \mathcal{A} can apply the oracle unitary U_f . This quantum oracle will sometimes also be denoted by \mathcal{O}_f .

Recall that a symmetric-key encryption scheme is a triple of classical probabilistic algorithms (KeyGen, Enc, 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 $\text{KeyGen}(1^n)$, then it holds that $\text{Dec}_k(\text{Enc}_k(m)) = m$ 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, it is implicit that \mathcal{A} is split into multiple QPTs (one for each round), and that these algorithms forward their internal (quantum) state to the next algorithm in the sequence.

2.2. Basic Number Theory Notation

We briefly review the following mathematical conventions. We denote the greatest common divisor (gcd) between two integers $a, b \in \mathbb{N}$ as $\text{gcd}(a, b)$. The multiplicative group of integers modulo a prime q is denoted as \mathbb{Z}_q^\times . Recall that for $m \in \mathbb{N}$, the Euler totient function $\varphi(m)$ counts the number of positive integers up to m that are relatively prime to m . The Möbius function $\mu(n)$ is defined as the sum of the primitive n -th roots of unity. Finally, we also recall that $\omega(m)$ denotes the prime omega function which counts the number of distinct prime factors of an integer $m \in \mathbb{N}$.

2.3. Quantum-Secure Pseudorandomness

A pseudorandom function is a family of deterministic and efficiently computable functions that appear random to any PPT adversary with adaptive (classical) oracle access. Similarly, a quantum-secure pseudorandom function achieves security against QPT adversaries with adaptive quantum oracle access. More specifically, let $f : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^\ell$ be an efficiently computable function, where n, m, ℓ are integers and where f defines a family of functions $\{f_k\}_{k \in \{0, 1\}^m}$ with $f_k(x) = f(k, x)$. We say f is a *quantum-secure pseudorandom function* (or QPRF) if, for every QPT \mathcal{A} ,

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

Here, \mathcal{F}_m^ℓ 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 [15,24,25].

A quantum-secure pseudorandom permutation is a bijective function family of quantum-secure pseudorandom functions. More specifically, consider a function $P : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^m$, where n and m are integers, such that each function $P_k(x) = P(k, x)$ in the corresponding family $\{P_k\}_{k \in \{0,1\}^n}$ is bijective. We say P is a *quantum-secure pseudorandom permutation* (or QPRP) if, for every QPT \mathcal{A} with access to both the function and its inverse,

$$\left| \Pr_{k \leftarrow \mathbb{S}_{\{0,1\}^n}} [\mathcal{A}^{P_k, P_k^{-1}}(1^n) = 1] - \Pr_{\pi \leftarrow \mathbb{S}_{\mathcal{P}_m}} [\mathcal{A}^{\pi, \pi^{-1}}(1^n) = 1] \right| \leq \text{negl}(n), \tag{2}$$

where \mathcal{P}_m denotes the set of permutations over m -bit strings. Throughout this work, we shall assume strong QPRPs under the above definition, i.e., such that the security is maintained despite additional access to an inverse. One can construct QPRPs from quantum-secure one-way functions [16].

2.4. 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 [26]:

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 $i \in \{1, \dots, N\}$, 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 require a bound on the number of qubits; 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 [27] (note that shared randomness *per se* does not allow them to communicate). Hence, Alice and Bob can pursue probabilistic strategies with access to the same random variable. Finally, the definition also takes into account that the encoder and decoder are allowed to share an entangled state as part of ρ_x .

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

2.5. Quantum Fourier Transform

For any positive integer q , the quantum Fourier transform over \mathbb{Z}_q is defined by the operation

$$\text{QFT}_{\mathbb{Z}_q} |x\rangle = \frac{1}{\sqrt{q}} \sum_{y \in \mathbb{Z}_q} \omega_q^{x \cdot y} |y\rangle,$$

where $\omega_q = e^{\frac{2\pi i}{q}}$. Due to early work by Kitaev [28], this variant of the Fourier transform can be implemented using quantum phase estimation in complexity polynomial in $\log q$. An improved approximate implementation of this operation is due to Hales and Hallgren [29].

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 $U_{Dec_k} : |c\rangle|m\rangle \mapsto |c\rangle|m \oplus Dec_k(c)\rangle$. 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 $\{U_{Enc_k,r}\}_r$ where

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

and r varies over all the 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 $U_{Enc_k,r}$ 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 3 (IND-QCCA1). Let $\Pi = (\text{KeyGen}, \text{Enc}, \text{Dec})$ be an encryption scheme, \mathcal{A} a QPT, and n the security parameter. Define $\text{IndGame}(\Pi, \mathcal{A}, n)$ as follows.

1. Setup: a key $k \leftarrow \text{KeyGen}(1^n)$ 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 $Enc_k(m_b)$ 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} ,

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

By inspection, one immediately sees that our definition lies between the established notions of IND-QCPA and IND-QCCA2 [10,12,14]. It will later be convenient to work with a variant of the game IndGame , which we now define.

Definition 4 ($\text{IndGame}'$). We define the experiment $\text{IndGame}'(\Pi, \mathcal{A}, n)$ just as $\text{IndGame}(\Pi, \mathcal{A}, n)$, except that in the pre-challenge phase \mathcal{A} only outputs a single message m , and in the challenge phase \mathcal{A} receives $Enc_k(m)$ if $b = 0$, and $Enc_k(x)$ for a uniformly random message x if $b = 1$.

Working with $\text{IndGame}'$ rather than IndGame does not change security. Specifically (as we show in Appendix A.2), Π is IND-QCCA1 if and only if, for every QPT \mathcal{A} , $\Pr[\neg \text{wins IndGame}'(\Pi, \mathcal{A}, n)] \leq 1/2 + \text{negl}(n)$.

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 $\mathcal{D}_{\text{Samp}}$, and h and f are functions with domain the support of $\mathcal{D}_{\text{Samp}}$. 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 5 (SEM-QCCA1). Let $\Pi = (\text{KeyGen}, \text{Enc}, \text{Dec})$ be an encryption scheme, and consider the experiment $\text{SemGame}(b)$ (with parameter $b \in \{\text{real}, \text{sim}\}$) with a QPT \mathcal{A} , defined as follows.

1. Setup: a key $k \leftarrow \text{KeyGen}(1^n)$ 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 $m \xleftarrow{\$} \text{Samp}$ is generated; \mathcal{A} receives $h(m)$ and gets access to an oracle for Enc_k only; if $b = \text{real}$, \mathcal{A} also receives $\text{Enc}_k(m)$; \mathcal{A} outputs a string s ;
4. Resolution: \mathcal{A} wins if $s = f(m)$.

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

$$|\Pr[\mathcal{A} \text{ wins SemGame}(\text{real})] - \Pr[\mathcal{S} \text{ wins SemGame}(\text{sim})]| \leq \text{negl}(n).$$

Our definition is a straightforward modification of SEM-QCPA [10,12]; The modification is to give \mathcal{A} and \mathcal{S} oracle access to Dec_k in the pre-challenge phase.

Theorem 6. Let $\Pi = (\text{KeyGen}, \text{Enc}, \text{Dec})$ 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., [30]) carries over directly to the quantum case. This was already observed for the case of QCPA by [10], and extends straightforwardly to the case where both the adversary and the simulator gain oracle access to Dec_k in the pre-challenge phase. 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.

4. Secure Constructions

4.1. PRF Scheme

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

Construction 1 (PRF scheme). Let n be the security parameter and let $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be an efficient family of functions $\{f_k\}_k$. The symmetric-key encryption scheme $\text{PRFscheme}[f] = (\text{KeyGen}, \text{Enc}, \text{Dec})$ is defined by the following efficient algorithms:

1. KeyGen: output a key $k \xleftarrow{\$} \{0,1\}^n$;
2. Enc: to encrypt a message \mathbf{m} , choose a random string $\mathbf{r} \xleftarrow{\$} \{0,1\}^n$ and output $(\mathbf{r}, f_k(\mathbf{r}) \oplus \mathbf{m})$;
3. Dec: to decrypt a ciphertext (\mathbf{r}, \mathbf{c}) , output $\mathbf{c} \oplus f_k(\mathbf{r})$;

For simplicity, we chose a particularly 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 below) are valid for arbitrary polynomial-size parameter choices.

We show that the above scheme satisfies QCCA1, provided that the underlying PRF is secure against quantum queries.

Theorem 7. If f is a QPRF, then $\text{PRFscheme}[f]$ is IND-QCCA1-secure.

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

We will work with the single-message variant of IndGame , $\text{IndGame}'$, described below as GAME 0. In Appendix A.2, we show that Π is IND-QCCA1 if and only if no QPT adversary can win $\text{IndGame}'$

with non-negligible bias. We first show that a version of $\text{IndGame}'$ where we replace f with a random function, called GAME 1 below, is indistinguishable from $\text{IndGame}'$, 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.

Game 0: This is the game $\text{IndGame}'(\Pi, \mathcal{A}, n)$, 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

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

Here, $\text{Enc}_k(x) := (r^*, f_k(r^*) \oplus x)$ 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 $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$.

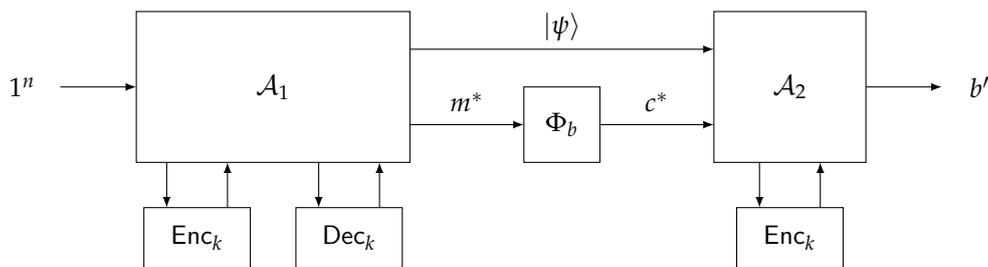


Figure 1. $\text{IndGame}'$ from Definition 4.

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 oracle distinguisher \mathcal{D} that distinguishes between the QPRF $\{f_k\}_k$ and a true random function, with distinguishing advantage

$$|\Pr[1 \leftarrow \text{GAME 0}] - \Pr[1 \leftarrow \text{GAME 1}]|,$$

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 $\text{IndGame}'$ as follows:

1. Run \mathcal{A}_1 , answering encryption queries using classical calls to g in place of f_k , and answering 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 $\delta_{bb'}$.

It remains to show that no QPT adversary can win GAME 1 with non-negligible probability. To do this, we design a quantum random access code from any adversary, and use the lower bound on the bias given in Lemma 1.

Intuition. We first give some intuition. In an encryption query, the adversary, either \mathcal{A}_1 or \mathcal{A}_2 , queries a message, or a 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 get a sample $(r, F(r))$. Thus, 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 , it follows there seem to be only two possible ways \mathcal{A}_2 could learn $F(r^*)$:

1. \mathcal{A}_2 gets lucky in one of his at most $\text{poly}(n)$ many queries to Enc_k and happens to sample the challenge pair $(r^*, F(r^*))$;
2. Or, the adversary 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 $\text{poly}(n)$ -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.

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).

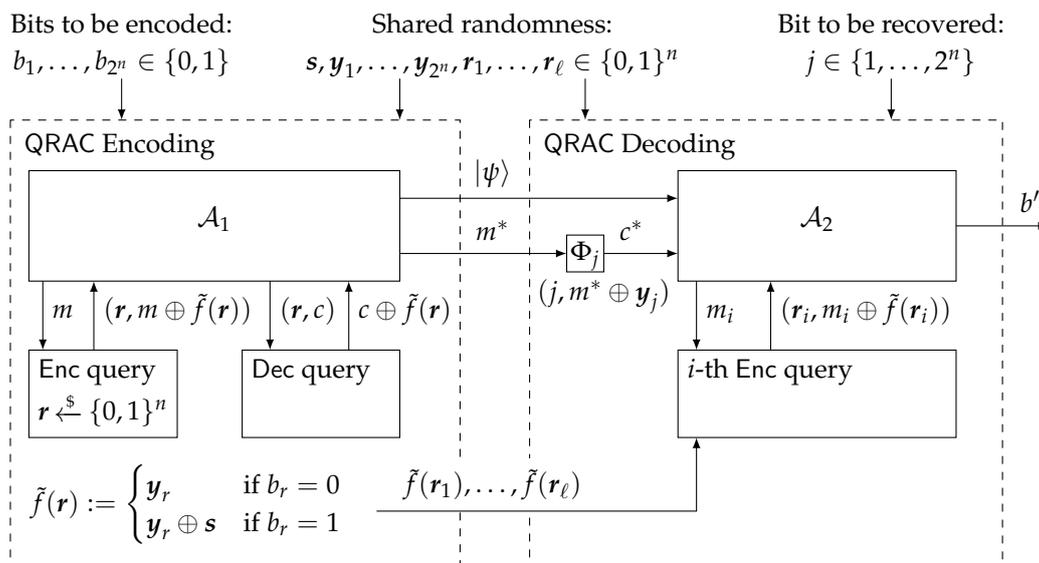


Figure 2. Quantum random access code construction for the pseudorandom function (PRF) scheme.

Encoding. Let $b_1, \dots, b_{2^n} \in \{0, 1\}$ be the string to be encoded. Let $s, y_1, \dots, y_{2^n} \in \{0, 1\}^n$ be given by the first $n(1 + 2^n)$ bits of the shared randomness, and let $r_1, \dots, r_\ell \in \{0, 1\}^n$ be the next

ℓn bits. Define a function $\tilde{f} : \{0,1\}^n \rightarrow \{0,1\}^n$ as follows. For $r \in \{0,1\}^n$, we will slightly abuse notation by letting r denote the corresponding integer value between 1 and 2^n . Define $\tilde{f}(r) = y_r \oplus b_r s$. 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 $\varrho = (|\psi\rangle, m^*, \tilde{f}(r_1), \dots, \tilde{f}(r_\ell))$.

Decoding. Let $j \in \{1, \dots, 2^n\}$ 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 $s, y_1, \dots, y_{2^n}, r_1, \dots, r_\ell$ given by the shared randomness. Upon receiving a query $j \in \{1, \dots, 2^n\}$, run \mathcal{A}_2 with inputs $|\psi\rangle$ and $(j, m^* \oplus y_j)$. On \mathcal{A}_2 's i -th encryption oracle call, use randomness r_i , so that if the input to the oracle is $|m, c\rangle$, the state returned is $|m, c \oplus (r_i, m \oplus \tilde{f}(r_i))\rangle$ (note that $\tilde{f}(r_i)$ 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 $b_1, \dots, b_{2^n} \in \{0,1\}$ and $j \in \{1, \dots, 2^n\}$, 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 $m^* \oplus y_j = m^* \oplus \tilde{f}(j)$, so the correct guess for \mathcal{A}_2 would be 0, and if $b_j = 1$, then $m^* \oplus y_j = m^* \oplus \tilde{f}(j) \oplus s = x \oplus \tilde{f}(j)$ for the uniformly random string $x = m^* \oplus s$, 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 $2^{\text{poly}(n)}$, since $|\psi\rangle$ must be a $\text{poly}(n)$ -qubit state (\mathcal{A}_1 only runs for $\text{poly}(n)$ time), and ℓ , the number of queries made by \mathcal{A}_2 must be $\text{poly}(n)$, since \mathcal{A}_2 only runs for $\text{poly}(n)$ time. As this code encodes 2^n bits into a state of dimension $2^{\text{poly}(n)}$, by Lemma 1 (proven in Appendix A.1), the bias is $O(2^{-n/2} \text{poly}(n)) = \text{negl}(n)$, so $p \leq \frac{1}{2} + \text{negl}(n)$. \square

4.2. PRP Scheme

We now prove the IND-QCCA1 security of a standard encryption scheme based on pseudorandom permutations.

Construction 2 (PRP scheme). Let n be the security parameter and let $P : \{0,1\}^n \times \{0,1\}^{2^n} \rightarrow \{0,1\}^{2^n}$ be an efficient family of permutations $\{P_k\}_k$. The symmetric-key scheme $\text{PRPscheme}[f] = (\text{KeyGen}, \text{Enc}, \text{Dec})$ is defined by the following efficient algorithms:

1. KeyGen: output $k \xleftarrow{\$} \{0,1\}^n$;
2. Enc: to encrypt $m \in \{0,1\}^n$, choose $r \xleftarrow{\$} \{0,1\}^{2^n}$ and output $P_k(m||r)$;
3. Dec: to decrypt $c \in \{0,1\}^{2^n}$, output the first n bits of $P_k^{-1}(c)$.

As before, we chose a simple set of parameters; in general, the randomness length, plaintext length, and security parameter can be related by arbitrary polynomials.

Theorem 8. If P is a QPRP, then $\text{PRPscheme}[P]$ is IND-QCCA1-secure.

Proof. We follow a similar proof strategy as with the PRF scheme. Fix a QPT adversary \mathcal{A} against $\Pi := \text{PRPscheme}[P] = (\text{KeyGen}, \text{Enc}, \text{Dec})$. We have that Π is IND-QCCA1 if and only if no QPT adversary can win $\text{IndGame}'$ with non-negligible bias. First, we show that a version of $\text{IndGame}'$ where we replace P with a random permutation, described below as GAME 1, is indistinguishable from $\text{IndGame}'$, 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.

Game 0: in the pre-challenge phase, \mathcal{A}_1 gets access to oracles Enc_k and Dec_k . in the challenge phase, \mathcal{A}_1 outputs m and its private data $|\psi\rangle$; a random bit $b \xleftarrow{\$} \{0, 1\}$ is sampled, and \mathcal{A}_2 is run on input $|\psi\rangle$ and a challenge ciphertext

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

where $r^* \xleftarrow{\$} \{0, 1\}^n$ and x is sampled uniformly at random. In the challenge phase, \mathcal{A}_2 has oracle access to Enc_k only and outputs a bit b' . The outcome of the game is simply the bit $\delta_{bb'}$.

Game 1: This is the same game as GAME 0, except we now replace P_k with a perfectly random permutation $\pi : \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$.

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 construct a quantum oracle distinguisher \mathcal{D} that distinguishes between P_k and a perfectly random permutation, with distinguishing advantage

$$|\Pr[1 \leftarrow \text{GAME 0}] - \Pr[1 \leftarrow \text{GAME 1}]|,$$

which must then be negligible since P_k is a QPRP. Here, the distinguisher \mathcal{D} receives quantum oracle access to a function φ , which is either P_k for a random k , or a random permutation π , and proceeds by simulating \mathcal{A} playing $\text{IndGame}'$ as follows:

1. Run \mathcal{A}_1 , answering encryption queries using oracle calls to φ in place of P_k , where for a given input and via randomness r ,

$$\text{Enc} : |m\rangle|c\rangle \mapsto |m\rangle|c \oplus \varphi(m||r)\rangle.$$

Answer decryption queries using quantum oracle calls to $\tilde{\varphi}^{-1}$, a function that first computes φ^{-1} but then (analogous to the PRP construction) discards the last n bits of the pre-image corresponding to the randomness, i.e.

$$\text{Dec} : |c\rangle|m\rangle \mapsto |c\rangle|m \oplus \tilde{\varphi}^{-1}(c)\rangle.$$

2. Simulate the challenge phase by sampling $b \xleftarrow{\$} \{0, 1\}$ and encrypting using a randomness r^* together with a classical call to φ in place of P_k ; run \mathcal{A}_2 and simulate encryption queries as before.
3. When \mathcal{A}_2 outputs b' , output $\delta_{bb'}$.

It remains to show that no QPT adversary can win GAME 1 with non-negligible probability. To do this, we will again design a random access code from any adversary's strategy with success probability p , and use the lower bound on the bias given in Lemma 1. We will then construct a QRAC with bias $\text{negl}(n)$ from this adversary, and hence conclude that $p \leq \frac{1}{2} + \text{negl}(n)$.

Construction of a quantum random access code. Let \mathcal{A} be a QPT adversary with winning probability p and let $\ell = \text{poly}(n)$ be an upper bound on the number of queries made by \mathcal{A}_2 . When constructing a QRAC for the PRP scheme, we shall also assume for simplicity that both the encoder and decoder share a random permutation (as part of the shared randomness). According to the well known *coupon collector's problem* [31], it is sufficient for the encoder and decoder to share around $N \ln(N)$ random strings on average, where N denotes the number of distinct random strings required to make up the desired permutation. We define a quantum random access code as follows (see also Figure 3).

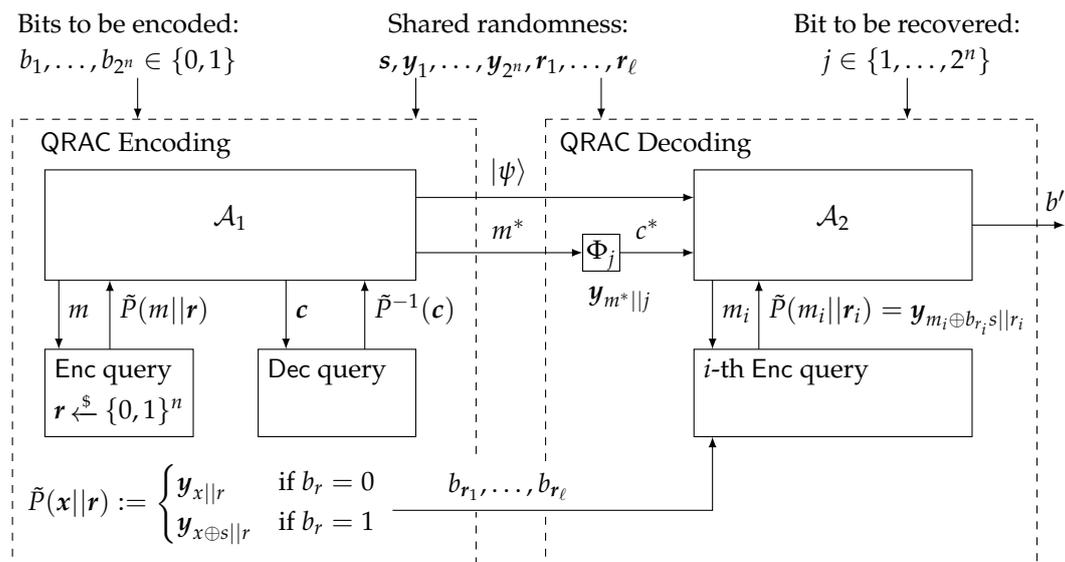


Figure 3. Quantum random access code construction for the pseudorandom permutation (PRP) scheme.

Encoding. Let $b_1, \dots, b_{2^n} \in \{0, 1\}$ be the string to be encoded and let the shared randomness be given by a random string s together with a random permutation $\mathbf{y} = \mathbf{y}_1, \dots, \mathbf{y}_{2^{2n}} \in \{0, 1\}^{2^n}$ and a set of random strings $\mathbf{r}_1, \dots, \mathbf{r}_\ell \in \{0, 1\}^n$. Using b_1, \dots, b_{2^n} , we define a new random permutation by letting $\tilde{P}(x||r) := \mathbf{y}_{x \oplus b_r s || r}$ (\tilde{P} remains a permutation since $\tilde{P}(x||r) = \tilde{P}(x'||r') \iff \mathbf{y}_{x \oplus b_r s || r} = \mathbf{y}_{x' \oplus b_{r'} s || r'} \iff (r = r') \wedge (x = x')$). Now run \mathcal{A}_1 by answering encryption and decryption queries using \tilde{P} in place of π (for decryption, use \tilde{P}^{-1} and discard the last n bits). Let m^* and $|\psi\rangle$ be the outputs of \mathcal{A}_1 . Then, output $\varrho = (|\psi\rangle, m^*, b_{r_1}, \dots, b_{r_\ell})$.

Decoding. Let $j \in \{1, \dots, 2^n\}$ be the index of the bit to be decoded; so given ϱ as above, we will recover b_j by making use of the shared randomness defined above. Upon receiving a query $j \in \{1, \dots, 2^n\}$, run \mathcal{A}_2 with inputs $|\psi\rangle$ and $c^* = \mathbf{y}_{m^* || j}$. 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 $b_1, \dots, b_{2^n} \in \{0, 1\}$ and $j \in \{1, \dots, 2^n\}$, 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{P} is a uniformly random permutation, and the queries are responded to just as in GAME 1. Further, note that if $b_j = 0$, the challenge amounts to $\tilde{P}(m^* || j) = \mathbf{y}_{m^* || j}$, so the correct guess for \mathcal{A}_2 would be 0, and if $b_j = 1$, then $\mathbf{y}_{x || j}$ is an encryption of a uniformly random string $x = m^* \oplus s$, so the correct guess for \mathcal{A}_2 would be 1.

Therefore, the average bias of the code is $p - 1/2$. We now proceed with a similar analysis as with the PRF scheme. Note that ϱ has dimension at most $2^{\text{poly}(n)}$, since $|\psi\rangle$ must be a $\text{poly}(n)$ -qubit state (\mathcal{A}_1 only runs for $\text{poly}(n)$ time), and ℓ , the number of queries made by \mathcal{A}_2 must be $\text{poly}(n)$, since \mathcal{A}_2 only runs for $\text{poly}(n)$ time. As this code encodes 2^n bits into a state of dimension $2^{\text{poly}(n)}$, by Lemma 1, the bias is $O(2^{-n/2} \text{poly}(n)) = \text{negl}(n)$, so $p \leq \frac{1}{2} + \text{negl}(n)$. \square

5. Quantum Algorithm for Linear Rounding Functions

In this section, we analyze the performance of the Bernstein–Vazirani algorithm [17] 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 $b \in \{1, \dots, q - 1\}$ (if b does not divide q , one of the blocks will have size $< b$).

Definition 6. 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 4):

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

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

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

In words, we divide the cyclic group \mathbb{Z}_q into c blocks, beginning at a : The first $c - 1$ have size b , and the last one has size $b - (cb - q) \leq b$ (see Figure 4). Then, the rounding function, LRF , classifies x by which of these blocks the value $\langle x, k \rangle$ falls into.

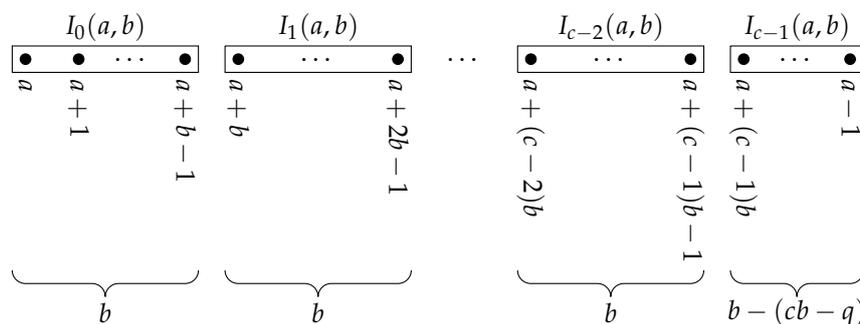


Figure 4. Dividing \mathbb{Z}_q into $c = \lceil q/b \rceil$ blocks, starting from a . The first $c - 1$ blocks, labelled $I_0(a, b), \dots, I_{c-2}(a, b)$, have size b and the last, labelled $I_{c-1}(a, b)$, contains the remaining $b - (cb - q) \leq b$ elements of \mathbb{Z}_q .

The following theorem shows that the modulo- q variant of the Bernstein–Vazirani algorithm (Algorithm 1) can recover k with constant probability of success by using only a single quantum query to $\text{LRF}_{k,a,b}$.

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 $U_{\text{LRF}} : |x\rangle|z\rangle \mapsto |x\rangle|z + \text{LRF}_{k,a,b}(x) \pmod{c}\rangle$ where $x \in \mathbb{Z}_q^n$, $z \in \mathbb{Z}_c$ and $\text{LRF}_{k,a,b}$ is the rounded inner product function for some unknown $k \in \mathbb{Z}_q^n$ and $a \in \mathbb{Z}_q$.

Output : String $\tilde{k} \in \mathbb{Z}_q^n$ such that $\tilde{k} = k$ 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 $\omega_c = e^{2\pi i/c}$:

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

2. Query the oracle U_{LRF} for $\text{LRF}_{k,a,b}$ to obtain $\frac{1}{\sqrt{q^n}} \sum_{x \in \mathbb{Z}_q^n} \omega_c^{-\text{LRF}_{k,a,b}(x)} |x\rangle \otimes \frac{1}{\sqrt{c}} \sum_{z=0}^{c-1} \omega_c^z |z\rangle$.

3. Discard the last register and apply the quantum Fourier transform $\text{QFT}_{\mathbb{Z}_q}^{\otimes n}$.

4. Measure in the computational basis and output the outcome \tilde{k} .
-

Theorem 9. Let U_{LRF} be the quantum oracle for the linear rounding function $\text{LRF}_{k,a,b}$ 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 $\mathbf{k} \in \mathbb{Z}_q^n$ such that \mathbf{k} has at least one entry that is a unit modulo q . By making one query to the oracle U_{LRF} , Algorithm 1 recovers the key \mathbf{k} with probability at least $(\frac{2}{\pi} - \frac{\pi}{8})^2 \approx 0.059$.

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 $c = \lceil q/b \rceil$ and $d = cb - q$. Throughout this proof, let $\text{LRF}(x) = \text{LRF}_{k,a,b}(x)$. 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:

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

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

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

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

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

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

$$|\langle \mathbf{k} | \psi \rangle|^2 = \left| \frac{1}{q^n} \sum_{x \in \mathbb{Z}_q^n} \omega_q^{-\frac{q}{c}\text{LRF}(x) + \langle x, \mathbf{k} \rangle} \right|^2 = \left| \frac{1}{q^n} \sum_{v=0}^{c-1} \omega_q^{-\frac{q}{c}v} \sum_{x \in \mathbb{Z}_q^n: \text{LRF}(x)=v} \omega_q^{\langle x, \mathbf{k} \rangle} \right|^2. \tag{3}$$

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

$$\begin{aligned} \sum_{x \in \mathbb{Z}_q^n: \text{LRF}(x)=v} \omega_q^{\langle x, \mathbf{k} \rangle} &= \sum_{x \in \mathbb{Z}_q^n: \langle x, \mathbf{k} \rangle \in I_v(a,b)} \omega_q^{\langle x, \mathbf{k} \rangle} \\ &= \sum_{\mathbf{y} \in \mathbb{Z}_q^{n-1}} \omega_q^{\langle \mathbf{y}, \mathbf{k}_{1:n-1} \rangle} \sum_{\substack{x_n \in \mathbb{Z}_q: \\ x_n k_n \in I_v(a - \langle \mathbf{y}, \mathbf{k}_{1:n-1} \rangle, b)}} \omega_q^{x_n k_n}. \end{aligned} \tag{4}$$

(recall the definition of $I_v(a, b)$ from Definition 6). Since k_n is a unit, for each $z \in I_v(a - \langle \mathbf{y}, \mathbf{k}_{1:n-1} \rangle)$, there is a unique $x_n \in \mathbb{Z}_q$ such that $x_n k_n = z$. Thus, for a fixed $\mathbf{y} \in \mathbb{Z}_q^{n-1}$, letting $a' = a - \langle \mathbf{y}, \mathbf{k}_{1:n-1} \rangle$, 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 (4) to get:

$$\sum_{\substack{x \in \mathbb{Z}_q^n \\ \text{LRF}(x)=v}} \omega_q^{\langle x, k \rangle} = \sum_{y \in \mathbb{Z}_q^{n-1}} \omega_q^{\langle y, k_{1:n-1} \rangle} \omega_q^{a - \langle y, k_{1:n-1} \rangle + vb} \sum_{z=0}^{b-1} \omega_q^z = q^{n-1} \omega_q^{a+vb} \sum_{z=0}^{b-1} \omega_q^z. \tag{5}$$

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

$$\sum_{x \in \mathbb{Z}_q^n : \text{LRF}(x)=c-1} \omega_q^{\langle x, k \rangle} = q^{n-1} \omega_q^{a-(b-d)} \sum_{z=0}^{b-d-1} \omega_q^z. \tag{6}$$

This is slightly different from the $v < c - 1$ case, shown in (5), but very similar. If we substitute $v = c - 1$ in (5) and compare it to (6), we get

$$\begin{aligned} & \left| q^{n-1} \omega_q^{a-(b-d)} \sum_{z=0}^{b-d-1} \omega_q^z - q^{n-1} \omega_q^{a-(b-d)} \sum_{z=0}^{b-1} \omega_q^z \right| \\ &= q^{n-1} \left| \sum_{z=b-d}^{b-1} \omega_q^z \right| = q^{n-1} \left| \sum_{z=0}^{d-1} \omega_q^z \right| = q^{n-1} \left| \frac{\sin(\pi d/q)}{\sin(\pi/q)} \right| \\ &\leq q^{n-1} \frac{\pi d/q}{2/q} = q^{n-1} \frac{\pi}{2} d. \end{aligned} \tag{7}$$

Above, we have used the facts $\sin x \leq x$, and $|\sin x| \geq 2x/\pi$ when $|x| \leq \pi/2$. Now, plugging (5) into (3) for all the $v < c - 1$ terms, and using (7) and the triangle inequality for the $v = c - 1$ term, we get:

$$\begin{aligned} |\langle k | \psi \rangle| &\geq \left| \frac{1}{q^n} \sum_{v=0}^{c-1} \omega_q^{-qv/c} \cdot q^{n-1} \omega_q^{a+vb} \sum_{z=0}^{b-1} \omega_q^z \right| - \left| \frac{1}{q^n} \omega_q^{-q(c-1)/c} \cdot q^{n-1} \frac{\pi}{2} d \right| \\ &= \frac{1}{q} \left| \sum_{v=0}^{c-1} \omega_q^{v(b-q/c)} \frac{\sin(b\pi/q)}{\sin(\pi/q)} \right| - \frac{\pi}{2} \frac{d}{q} \\ &= \frac{1}{q} \frac{\sin(b\pi/q)}{\sin(\pi/q)} \left| \sum_{v=0}^{c-1} \omega_q^{v(b-q/c)} \right| - \frac{\pi}{2} \frac{d}{q}. \end{aligned} \tag{8}$$

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

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

$$\geq c \left| \sqrt{1 - (2\pi d/q)^2} \right| \geq c \sqrt{\frac{\pi^2}{4} - 1}. \tag{10}$$

To get the inequalities (9) and (10) and , 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 $\frac{2\pi v d}{c q} \leq \frac{\pi}{2}$. 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

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

Since $\sin(\pi/q) \leq \pi/q$ and $q \geq 2b$, and by the assumption that $d/q \leq 1/4$, we get

$$\frac{\sin(b\pi/q)}{\sin(\pi/q)} \geq \frac{\sqrt{1 - \left(\frac{\pi d}{2q} \right)^2}}{\pi/q} \geq \frac{2b}{\pi} \sqrt{1 - \frac{\pi^2}{64}}.$$

Thus, in both cases,

$$\frac{\sin(b\pi/q)}{\sin(\pi/q)} \geq \frac{2b}{\pi} \sqrt{1 - \frac{\pi^2}{64}}.$$

Plugging this and (10) into (8), and again using the assumption $d/q \leq 1/4$, we get:

$$|\langle \mathbf{k}, \psi \rangle| \geq \frac{1}{q} \frac{\sin(b\pi/q)}{\sin(\pi/q)} \left| \sum_{v=0}^{c-1} \omega_q^{v(b-q/c)} \right| - \frac{\pi d}{2q} \tag{11}$$

$$\begin{aligned} &\geq \frac{2bc}{\pi q} \sqrt{\left(1 - \frac{\pi^2}{64}\right) \cdot \left(\frac{\pi^2}{4} - 1\right)} - \frac{\pi}{8} \\ &\geq \frac{2bc}{\pi q} - \frac{\pi}{8} = \frac{2q+d}{\pi} - \frac{\pi}{8} \\ &\geq \frac{2}{\pi} - \frac{\pi}{8}, \end{aligned} \tag{12}$$

completing the proof. \square

6. Key Recovery Against LWE

In this section, we consider various LWE-based encryption schemes and show using Theorem 9 that the decryption key can be efficiently recovered using a single quantum decryption query (Sections 6.1–6.3). Then, in Section 6.4, we 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 [3].

Construction 3 (LWE-SKE [3]). *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 LWE-SKE(n, q, χ) = (KeyGen, Enc, Dec) is defined as follows:*

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

We refer to an encryption scheme as *correct* if $\Pr[\text{Dec}_k(\text{Enc}_k(b)) = b] = 1 - \text{negl}(n)$. To guarantee that the above LWE-SKE scheme is correct, we need to restrict the support of the discrete error distribution χ by fixing the noise magnitude η so that $|e| \leq \lfloor \frac{q}{4} \rfloor$ for all $e \leftarrow \chi$.

As a corollary of Theorem 9, an adversary that is granted a single quantum decryption query can recover the key with probability at least $(\frac{2}{\pi} - \frac{\pi}{8})^2$:

Corollary 1. *There is a quantum algorithm that makes one quantum query to LWE-SKE.Dec_k and recovers the entire key k with probability at least $(\frac{2}{\pi} - \frac{\pi}{8})^2 \approx 0.059$.*

Proof. Note that LWE-SKE.Dec_k coincides with a linear rounding function $\text{LRF}_{k',a,b}$ for a key $k' = (-k, 1) \in \mathbb{Z}_q^{n+1}$, which has a unit in its last entry. We consider the following two cases: if q is not 3 mod 4, set $b = \lceil q/2 \rceil$, $a = -\lfloor q/4 \rfloor$, hence we get $\text{Dec}_k = \text{LRF}_{k',a,b}$. If $q = 3 \pmod 4$, set $b = \lceil q/2 \rceil$, $a = \lceil q/4 \rceil$, and then we have $\text{Dec}_k = 1 - \text{LRF}_{k',a,b}$. Our claim follows in both cases. Thus, by Theorem 9, Algorithm 1 makes one quantum query to $\text{LRF}_{k',a,b}$, which can be implemented using one quantum query to LWE-SKE.Dec_k , and recovers k' , and thus k , with probability at least $(\frac{2}{\pi} - \frac{\pi}{8})^2$. \square

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 $2^{t-n \log q}$. Using a linear number of classical queries, a straightforward key-recovery algorithm does in fact recover the key with constant success probability. One can simply query “unit vectors” of the form $e^{(i)} = (0, \dots, 0, 1, 0, \dots, 0)$ in order to recover one entry of the key at a time by averaging over errors.

6.2. Key Recovery via One Decryption Query in Public-Key LWE

The key-recovery attack described in Corollary 3 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 [3]. The second is the IND-CPA-secure public-key encryption scheme FrodoPKE [18], which is based on a construction of Lindner and Peikert [19]. 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 4 (LWE-PKE [3]). *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 . The public-key scheme $\text{LWE-PKE}(n, q, \chi) = (\text{KeyGen}, \text{Enc}, \text{Dec})$ is defined by the following efficient algorithms:*

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

Corollary 2. *There is a quantum algorithm that makes one quantum query to LWE-PKE.Dec_{sk} and recovers the entire key sk with probability at least $(\frac{2}{\pi} - \frac{\pi}{8})^2 \approx 0.059$.*

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 [19]. Compared to LWE-PKE, this scheme significantly reduces the key-size and achieves better security estimates than the initial proposal by Regev [3]. For a detailed discussion of FrodoPKE, we refer to [18]. We present the entire scheme for completeness, but the important part for our purposes is the decryption procedure.

Construction 5 (FrodoPKE [18]). *Let n, \bar{m}, \bar{n} be integer parameters, let $q \geq 2$ be 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 $\text{FrodoPKE} = (\text{KeyGen}, \text{Enc}, \text{Dec})$ is defined as follows:*

1. **KeyGen:** generate a matrix $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^{n \times n}$ and matrices $\mathbf{S}, \mathbf{E} \xleftarrow{\chi} \mathbb{Z}_q^{n \times \bar{n}}$; compute $\mathbf{B} = \mathbf{a}\mathbf{S} + \mathbf{E} \in \mathbb{Z}_q^{n \times \bar{n}}$; output the key-pair $(\mathbf{pk}, \mathbf{sk})$ with public key $\mathbf{pk} = (\mathbf{A}, \mathbf{B})$ and secret key $\mathbf{sk} = \mathbf{S}$.
2. **Enc:** to encrypt $\mathbf{m} \in \{0, 1\}^{B \cdot \bar{m} \cdot \bar{n}}$ (encoded as a matrix $\mathbf{M} \in \mathbb{Z}_q^{\bar{m} \times \bar{n}}$ with each entry having 0s in all but the B most significant bits) with public key \mathbf{pk} , sample error matrices $\mathbf{S}', \mathbf{E}' \xleftarrow{\chi} \mathbb{Z}_q^{\bar{m} \times n}$ and $\mathbf{E}'' \xleftarrow{\chi} \mathbb{Z}_q^{\bar{m} \times \bar{n}}$; compute $\mathbf{C}_1 = \mathbf{S}'\mathbf{a} + \mathbf{E}' \in \mathbb{Z}_q^{\bar{m} \times n}$ and $\mathbf{C}_2 = \mathbf{M} + \mathbf{S}'\mathbf{B} + \mathbf{E}'' \in \mathbb{Z}_q^{\bar{m} \times \bar{n}}$; output the ciphertext $(\mathbf{C}_1, \mathbf{C}_2)$.
3. **Dec:** to decrypt $(\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\bar{m} \times n} \times \mathbb{Z}_q^{\bar{m} \times \bar{n}}$ with secret-key $\mathbf{sk} = \mathbf{S}$, compute $\mathbf{M} = \mathbf{C}_2 - \mathbf{C}_1\mathbf{S} \in \mathbb{Z}_q^{\bar{m} \times \bar{n}}$. 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 \mathbf{S} using a single quantum query to FrodoPKE.Dec_S . If $\bar{m} = \bar{n}$, as in sample parameters given in [18], then this algorithm recovers \mathbf{S} completely.

Theorem 10. *There exists a quantum algorithm that makes one quantum query to FrodoPKE.Dec_S and recovers any choice of \bar{m} of the \bar{n} columns of \mathbf{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 $(\frac{2}{\pi} - \frac{\pi}{8})^2 \approx 0.059$.*

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

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

for any $c \in \mathbb{Z}_q^n$ and $z_1, \dots, z_{\bar{n}} \in \mathbb{Z}_{2^B}$. We first argue that one call to FrodoKEM.Dec_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 \mathbf{S} with probability $(\frac{2}{\pi} - \frac{\pi}{8})^2 \approx 0.059$, as long as it has at least one entry that is odd.

Let $\text{Trunc} : \mathbb{Z}_q \mapsto \mathbb{Z}_{2^B}$ 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 $\mathbf{C}_1 \in \mathbb{Z}_q^{\bar{m} \times n}$, $\mathbf{C}_2 = 0^{\bar{m} \times \bar{n}}$, and any $\{z_{i,j}\}_{i \in [\bar{m}], j \in [\bar{n}]} \subseteq \mathbb{Z}_{2^B}$:

$$U_{\text{FrodoKEM.Dec}} : |\mathbf{C}_1\rangle|0^{\bar{m} \cdot \bar{n}}\rangle \bigotimes_{i \in [\bar{m}], j \in [\bar{n}]} |z_{i,j}\rangle \mapsto |\mathbf{C}_1\rangle|0^{\bar{m} \cdot \bar{n}}\rangle \bigotimes_{i \in [\bar{m}], j \in [\bar{n}]} |z_{i,j} + \text{Trunc}([\mathbf{C}_1\mathbf{S}]_{i,j})\rangle. \quad (13)$$

Above, $[\mathbf{C}_1\mathbf{S}]_{i,j}$ represents the ij -th entry of $\mathbf{C}_1\mathbf{S}$. If $c^1, \dots, c^{\bar{m}}$ denote the rows of \mathbf{C}_1 , then $[\mathbf{C}_1\mathbf{S}]_{i,j} = \langle c^i, s^j \rangle$. Thus, $\text{Trunc}([\mathbf{C}_1\mathbf{S}]_{i,j}) = \text{LRF}_{s^j, 0, q/2^B}(c^i)$, 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 $C_2 = 0$), the right-hand side of (13) becomes

$$|c^1\rangle \dots |c^{\bar{m}}\rangle \bigotimes_{i \in [\bar{m}], j \in [\bar{n}]} |z_{i,j} + \text{LRF}_{s^j, 0, q/2^B}(c^i)\rangle. \tag{14}$$

Reordering the registers of (14), we get:

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

Thus, we can implement $U^{\otimes \bar{m}}$ using a single call to FrodoKEM.Dec_S.

Next we show that for any particular $j \in [\bar{n}]$, a single call to U can be used to recover s^j , the j -th column of S , with probability at least $(\frac{2}{\pi} - \frac{\pi}{8})^2 \approx 0.059$, as long as at least one entry of s^j is odd. To do this, we show how one use of U can be used to implement one phase query to $\text{LRF}_{s^j, 0, q/2^B}$. Then the result follows from the proof of Theorem 9.

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 $c \in \mathbb{Z}_q^n$, we have:

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

since addition here is modulo 2^B , and

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

Thus:

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

Thus, by the proof of Theorem 9, if we apply U to $q^{-n/2} \sum_{c \in \mathbb{Z}_q^n} |c\rangle|\phi_j\rangle$, Fourier transform the first register, and then measure, assuming s^j has at least one entry that is a unit (since q is a power of 2, this is just an odd number) we will measure s^j with probability at least $(\frac{2}{\pi} - \frac{\pi}{8})^2 \approx 0.059$.

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

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

Fourier transform each of the c registers, and then measure. \square

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 [20,21]. 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.

In order to generate samples, we assume a symmetric error distribution χ that samples “small” error polynomials from \mathcal{R} . For example, a typical choice (under a particular representation for \mathcal{R}_q) is to use an n -dimensional Gaussian, more specifically the product of n centered one-dimensional Gaussian distributions.

Let us first recall the basic public-key encryption scheme based on Ring-LWE. We restrict our analysis to single-bit encryption only.

Construction 6 (Ring-LWE-PKE [20,21]). *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. To speed up multiplication of large polynomials, various embeddings are adopted in practice that represent elements in \mathcal{R}_q as vectors in \mathbb{Z}_q^n [20]. For example, one can adopt the number-theoretic transform (NTT) to reduce polynomial multiplication to the much faster component-wise multiplication, such as in the post-quantum proposal *New Hope* [32]. We emphasize that our results are independent of the actual embedding used in practice, as long as the representation is an isomorphism between \mathcal{R}_q and \mathbb{Z}_q^n . The same is true for the choice of ring \mathcal{R} under a cyclotomic polynomial, and ultimately results in only slightly different classical post-processing. We show the following corollary to Theorem 9.

Corollary 3. *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 $(\frac{2}{\pi} - \frac{\pi}{8})^2 \approx 0.059$.*

Proof. Let us first analyze the decryption function. We let $(p)_0$ denote the constant term of a polynomial $p \in \mathcal{R}_q$. Then, for any two polynomials $u = \sum_{j=0}^{n-1} u_j X^j$ and $s = \sum_{j=0}^{n-1} s_j X^j \in \mathcal{R}_q$, we can identify the constant term of $u \cdot s$ as

$$\begin{aligned} (u \cdot s)_0 &= u_0 s_0 + \sum_{j=1}^{n-1} u_j s_{n-j} X^j X^{n-j} \\ &\equiv u_0 s_0 - u_1 s_{n-1} - u_2 s_{n-2} - \dots - u_{n-1} s_1 \pmod{q}, \end{aligned} \tag{15}$$

since $X^n \equiv -1$ in \mathcal{R}_q . We show that the outcome of Ring-LWE-PKE.Dec_s(u, v) coincides with a binary linear rounding function over \mathbb{Z}_q^n . Let $\mathbf{u}, \mathbf{s} \in \mathbb{Z}_q^n$ denote the coefficient vectors of $u, s \in \mathcal{R}_q$ respectively, and define a constant polynomial $v \equiv v_0 \in \mathcal{R}_q$ and vector $\mathbf{u}' := (\mathbf{u}, v_0) \in \mathbb{Z}_q^{n+1}$, for some $v_0 \in \mathbb{Z}_q$. We find that Ring-LWE-PKE.Dec_s(u, v_0) rounds the inner product $\langle \mathbf{u}', \mathbf{s}' \rangle$, where

$$s' = (-s_0 \pmod q, s_{n-1}, \dots, s_1, 1).$$

Thus, we can run the Bernstein–Vazirani algorithm for binary linear rounding functions on a uniform superposition over \mathbb{Z}_q^n and recover s from s' with simple classical post-processing. Note also that any choice of isomorphism between \mathcal{R}_q and \mathbb{Z}_q^n necessarily preserves the inner product in Equation (15), and hence any measurement outcome can be mapped back to the standard basis prior to post-processing—independently of the actual ring representation used in practice.

By Theorem 9, Algorithm 1 makes one quantum query to $\text{LRF}_{s',q}$, which can be implemented using one quantum query to $\text{Ring-LWE-PKE.Dec}_s$, and recovers the string s' , and thus also s , with probability $(\frac{2}{\pi} - \frac{\pi}{8})^2$. \square

6.4. Key Recovery via a Randomness-Access Query

While a linear number of classical decryption queries can be used to break LWE-based schemes, we have shown that only a single *quantum* decryption query is required. A natural question to ask is whether a similar statement can be made for *encryption* queries. Classically, it is known that the symmetric key version of LWE described in Construction 3, LWE-SKE, can be broken using a linear number of classical encryption queries when the adversary is also allowed to choose the randomness used by the query: The adversary simply sets the value to $e = 0$ each time, with a taking n linearly independent values.

In case the adversary is allowed to make quantum encryption queries with randomness access, a *single* quantum query suffice to recover *the entire key* with non-negligible probability, even when the adversary only has control over *a part* of the randomness used by the encryption: The randomness used to prepare vectors a , but not the randomness used to select the error e . Specifically, the adversary is given quantum oracle access to the *randomness-access encryption oracle* $U_{\text{Enc}_k}^{\text{RA}}$ such that, on input $(b; a)$, the adversary receives

$$\text{Enc}_k^{\text{RA}}(b; a) = (a, \langle a, k \rangle + b \lfloor q/2 \rfloor + e),$$

where $e \leftarrow \chi$. We extend this to a quantum randomness-access oracle by answering each element of the superposition using i.i.d. errors $e_a \leftarrow \chi$:

$$U_{\text{Enc}_k}^{\text{RA}} : |m\rangle|a\rangle|c\rangle \mapsto |m\rangle|a\rangle|c \oplus \text{Enc}_k^{\text{RA}}(m; a)\rangle.$$

This model is identical to the noisy learning model considered by Grilo et al. [22] which was inspired by the original noise model of Bshouty and Jackson [33]. First, it is not hard to see that algorithms making classical queries to the above oracle can extract at most $\log q$ bits of key from each query (specifically, from the last component of the ciphertext), and thus still require a linear number of queries to recover the complete key with non-negligible probability.

On the other hand, by a slight generalization of the proof of Theorem IV.1 from Ref. [22], we can recover the entire key with inverse polynomial success probability using a single query to $U_{\text{Enc}_k}^{\text{RA}}$, as long as the noise magnitude η is polynomial in n . While Grilo et al. show how to apply Bernstein–Vazirani to solve a certain *computational learning* version of LWE with quantum samples, we adopt a wider set of cryptographic security parameters and show that this algorithm leads to a dramatic quantum speed-up in an actual attack. Specifically, our improved analysis shows that the modulus need not be prime. In particular, in the case of only a single query to a quantum randomness-access encryption oracle, the entire key can be recovered with inverse polynomial success probability, as long as the noise magnitude η is polynomial in n . This is certainly the case for the vast majority of schemes in practice. Recall that φ denotes the Euler totient function. Specifically, for $m \in \mathbb{N}$, $\varphi(m)$ counts the number of integers up to m that are relative prime to m . If m is prime, then $\varphi(m) = m - 1$. Our algorithm to recover the key is as follows.

Theorem 11. Consider $\text{LWE-SKE}(n, q, \chi)$ with an arbitrary integer modulus $2 \leq q \leq \exp(n)$ and a symmetric error distribution χ of noise magnitude η . Then, Algorithm 2 makes one query to a randomness-accessible quantum encryption oracle for $\text{LWE-SKE}(n, q, \chi)$ and recovers the entire key with probability at least $\varphi(q)/(24\eta q) - o(1)$.

Proof. We follow a similar analysis as in Theorem IV.1 in [22]. Let k denote the key as sampled by $\text{LWE-SKE}(n, q, \chi)$. After querying $U_{\text{Enc}_k}^{\text{RA}}$ upon a state corresponding to an encryption of a message $b_0 = 0$, together with a choice of uniform superposition over the randomness register,

$$\frac{1}{\sqrt{q^n}} \sum_{x \in \mathbb{Z}_q^n} |b_0\rangle |x_1\rangle \dots |x_n\rangle |0\rangle,$$

Algorithm 2: Bernstein–Vazirani for randomness-accessible LWE encryption

input : Quantum randomness-access oracle $U_{\text{Enc}_k}^{\text{RA}}$ for $\text{LWE-SKE}(n, q, \chi)$

output: $k \in \mathbb{Z}_q^n$ with probability $\approx \varphi(q)/(24\eta q)$

1. Prepare a state corresponding to the message $b_0 = 0$ and uniform randomness:

$$\frac{1}{\sqrt{q^n}} \sum_{x \in \mathbb{Z}_q^n} |b_0\rangle |x_1\rangle \dots |x_n\rangle |0\rangle.$$

2. Query the randomness-accessible oracle $U_{\text{Enc}_k}^{\text{RA}}$, resulting in

$$\frac{1}{\sqrt{q^n}} \sum_{x \in \mathbb{Z}_q^n} |b_0\rangle |x_1\rangle \dots |x_n\rangle |\langle x, k \rangle + b_0 \lfloor q/2 \rfloor + e_x \pmod{q}\rangle.$$

3. Discard the first register and apply the quantum Fourier transform $\text{QFT}_{\mathbb{Z}_q}^{\otimes n+1}$.
 4. Measure in the computational basis, yielding an outcome $|y_1\rangle \dots |y_n\rangle |z\rangle$.
 5. If $\text{gcd}(z, q) = 1$, output $\tilde{k} = -(y_1, \dots, y_n)/z \pmod{q}$,
else output \perp .
-

Algorithm 2 receives a sample $|\psi\rangle$ where, for unknown independent errors $e_x \leftarrow \chi$ of noise magnitude $|e_x| \leq \eta$,

$$|\psi\rangle = \frac{1}{\sqrt{q^n}} \sum_{x \in \mathbb{Z}_q^n} |b_0\rangle |x_1\rangle \dots |x_n\rangle |\langle x, k \rangle + e_x \pmod{q}\rangle.$$

Thus, when discarding the first register and applying the quantum Fourier transform, we find

$$\text{QFT}_{\mathbb{Z}_q}^{\otimes n+1} |\psi\rangle = \frac{1}{q^{n+1/2}} \sum_{x \in \mathbb{Z}_q^n} \sum_{y \in \mathbb{Z}_q^n} \sum_{z \in \mathbb{Z}_q} \omega^{\langle x, y+z k \rangle + e_x z} |y_1\rangle \dots |y_n\rangle |z\rangle.$$

When measuring in the standard basis, the probability of obtaining outcome y_1, \dots, y_n, z is

$$\frac{1}{q^{2n+1}} \left| \sum_{x \in \mathbb{Z}_q^n} \omega^{\langle x, y+z k \rangle + e_x z} \right|^2.$$

In particular, the probability that $\mathbf{y} = -z\mathbf{k} \pmod{q}$ and also that z is prime relative to q (i.e., $\gcd(z, q) = 1$ and thus $z \in \mathbb{Z}_q^\times$) is given by

$$\begin{aligned} \frac{1}{q^{2n+1}} \sum_{z \in \mathbb{Z}_q^\times} \left| \sum_{x \in \mathbb{Z}_q^n} \omega^{e_x z} \right|^2 &= \frac{1}{q^{2n+1}} \sum_{z \in \mathbb{Z}_q^\times} \left[\left(\sum_{x \in \mathbb{Z}_q^n} \operatorname{Re}(\omega^{e_x z}) \right)^2 + \left(\sum_{x \in \mathbb{Z}_q^n} \operatorname{Im}(\omega^{e_x z}) \right)^2 \right] \\ &\geq \frac{1}{q^{2n+1}} \sum_{z \in \mathbb{Z}_q^\times} \left(\sum_{x \in \mathbb{Z}_q^n} \cos\left(\frac{2\pi e_x z}{q}\right) \right)^2 \geq \frac{1}{q^{2n+1}} \sum_{\substack{z \in \mathbb{Z}_q^\times \\ 1 \leq z \leq \frac{q}{6\eta}}} \left(\sum_{x \in \mathbb{Z}_q^n} \frac{1}{2} \right)^2 \\ &\geq \frac{1}{4q} \sum_{\substack{z \in \mathbb{Z}_q^\times \\ 1 \leq z \leq \lfloor \frac{q}{6\eta} \rfloor}} 1 \geq \frac{\varphi(q)}{24\eta q} - O(1/q). \end{aligned}$$

in the last inequality, we used the fact that $\lfloor \frac{q}{6\eta} \rfloor = \frac{q}{6\eta} + O(1)$, as well as a standard identity on the Euler totient function which we state in Lemma A2: Let $n, m > 1$ be integers and let ω be the prime omega function. Then, the following identity holds:

$$\sum_{\substack{1 \leq j \leq n \\ \gcd(j, m) = 1}} 1 = n \frac{\varphi(m)}{m} + O\left(2^{\omega(m)}\right).$$

Therefore, Algorithm 2 outputs the correct key $\tilde{\mathbf{k}}$ with probability at least $\varphi(q)/(24\eta q) - o(1)$. Note that, because $\varphi(q) = \Omega(q/\log \log(q))$, the success probability is inverse polynomial so long as the noise magnitude η is polynomial in n . \square

For completeness, we also consider a slightly different and yet plausible error model for quantum randomness-access queries. Unlike the previous model of i.i.d. errors, we now consider a single error $e \leftarrow \chi$ being sampled for every branch of the superposition of a single query. Our algorithm will crucially exploit the property that the error does not depend on any of the particular vectors in the respective branches of the wave function.

Surprisingly, we find that the error model of choice does matter, and greatly impacts the analysis of the algorithm. In fact, as we show next, one can now recover \mathbf{k} with certainty using only a single query to the randomness access encryption oracle. We adapt Algorithm 2 as follows:

We are left with the following Corollary.

Corollary 4. Consider $\text{LWE-SKE}(n, q, \chi)$ with an arbitrary integer modulus $2 \leq q \leq \exp(n)$ and a symmetric error distribution χ of noise magnitude η . Then, there exists a quantum algorithm that makes one query to a single-error randomness-accessible quantum encryption oracle for $\text{LWE-SKE}(n, q, \chi)$ and recovers the entire key with probability 1.

Proof. The result follows straight from the definition of the quantum Fourier transform. After applying the transformation in step 3 of Algorithm 3, the register contains the entries (k_1, \dots, k_n) . \square

Algorithm 3: Bernstein–Vazirani for (single-error) randomness-accessible LWE encryption

input : Quantum randomness-access oracle $U_{\text{Enc}_k}^{\text{RA}}$ for LWE-SKE(n, q, χ) in the single-error model.
output: $k \in \mathbb{Z}_q^n$ with probability 1

1. Prepare a particular state corresponding to the message $b_0 = 0$ as follows:

$$\frac{1}{\sqrt{q^n}} \sum_{x \in \mathbb{Z}_q^n} |b_0\rangle |x_1\rangle \cdots |x_n\rangle \otimes \frac{1}{\sqrt{q}} \sum_{z=0}^{q-1} \omega_q^z |z\rangle.$$

2. Query the randomness-accessible oracle $U_{\text{Enc}_k}^{\text{RA}}$ with single error $e \leftarrow \chi$:

$$\frac{1}{\sqrt{q^n}} \sum_{x \in \mathbb{Z}_q^n} \omega_q^{-\langle x, k \rangle} |b_0\rangle |x_1\rangle \cdots |x_n\rangle \otimes \frac{1}{\sqrt{q}} \sum_{z=0}^{q-1} \omega_q^z |z + e\rangle.$$

3. Discard the first and last register, and apply the quantum Fourier transform $\text{QFT}_{\mathbb{Z}_q}^{\otimes n}$.
 4. Measure in the computational basis to obtain $k \in \mathbb{Z}_q^n$.
-

Author Contributions: All authors contributed equally to this work. Conceptualization, G.A., S.J., M.O. and A.P.; formal analysis, G.A., S.J., M.O. and A.P.; writing—original draft preparation, G.A., S.J., M.O. and A.P.; writing—review and editing, G.A., S.J., M.O. and A.P. All authors have read and agreed to the published version of the manuscript.

Funding: GA is supported by National Science Foundation (NSF) grant CCF-1763736. SJ is supported by an NWO WISE Grant and NWO Veni Innovational Research Grant under project number 639.021.752. MO is supported by Leverhulme Trust Early Career Fellowship (ECF-2015-256). AP is partially supported by AFOSR YIP award number FA9550-16-1-0495, the IQIM—an NSF Physics Frontiers Center (NSF Grant PHY-1733907), and the Kortschak Scholars program.

Acknowledgments: We thank Ronald de Wolf for helpful discussions, Jop Briët for Lemma A1 and Peter Humphries for providing us with a simple proof of Lemma A2.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A.

Appendix A.1. Bound for Quantum Random Access Codes

Recall that a *quantum random access code* (QRAC) is the following scenario involving two parties, Alice and Bob [26]:

- Alice receives an N -bit string x and encodes it as a quantum state ρ_x .
- Bob receives ρ_x from Alice and is asked to recover the i -th bit of x , for some $i \in \{1, \dots, N\}$, by measuring the state.
- They win if Bob’s output agrees with x_i and lose otherwise.

A variation of this scenario allows Alice and Bob to use *shared randomness* in their encoding and decoding operations [27] (note that shared randomness *per se* does not allow them to communicate). If we denote the shared random variable by λ , Alice’s can now produce ρ_x^λ

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 A1. *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 $\rho_x^\lambda \geq 0$, $\text{Tr} \rho_x^\lambda = 1$, 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 $(-1)^{x_i} \text{Tr}(\rho_x^\lambda M_i^\lambda) / 2$. Moreover, if the average bias of the code is ϵ , then $\mathbb{E}_\lambda \mathbb{E}_{x,i} (-1)^{x_i} \text{Tr}(\rho_x^\lambda M_i^\lambda) \geq 2\epsilon$. We can rearrange this expression and upper bound each term using its operator norm, and then apply the noncommutative Khintchine inequality [34]:

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

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. \square

Appendix A.2. Equivalence of QCCA Models

Recall that the IND-QCCA1 notion is based on the security game IndGame defined in Definition 3. In the alternative security game $\text{IndGame}'$ (see Definition 4), 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 A1. *An encryption scheme Π is IND-QCCA1 if and only if for every QPT \mathcal{A} ,*

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

Proof. Fix a scheme Π . For one direction, suppose Π is IND-QCCA1 and let \mathcal{A} be an adversary against $\text{IndGame}'$. Define an adversary \mathcal{A}_0 against 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 $\text{IndGame}'(\Pi, \mathcal{A}, n)$ is then identical to $\text{IndGame}(\Pi, \mathcal{A}_0, n)$, which in turn must be negligibly close to uniform by IND-QCCA1 security of Π .

For the other direction, suppose no adversary can win $\text{IndGame}'$ with probability better than $1/2$, and let \mathcal{B} be an adversary against IndGame . Now, define two adversaries \mathcal{B}_0 and \mathcal{B}_1 against $\text{IndGame}'$ 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

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

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

Lemma A2. *Let $n, m > 1$ be integers. Then, the following holds where $\omega(m)$ is the prime omega function:*

$$\sum_{\substack{1 \leq j \leq n \\ \gcd(j,m)=1}} 1 = n \frac{\varphi(m)}{m} + O(2^{\omega(m)}).$$

For completeness, we include a short and simple proof of the above statement.

Proof ([35]).

$$\sum_{\substack{j=1 \\ \gcd(j,m)=1}}^n 1 = \sum_{j=1}^n \sum_{d|\gcd(j,m)} \mu(d) = \sum_{d|m} \mu(d) \sum_{\substack{j=1 \\ j \equiv 0 \pmod{d}}}^n 1$$

Here, μ denotes the Möbius function. The sum over $j = 1, \dots, n$ is equal to $\lfloor \frac{n}{d} \rfloor = \frac{n}{d} + O(1)$. Therefore,

$$n \sum_{d|m} \frac{\mu(d)}{d} + O\left(\sum_{d|m} |\mu(d)|\right) = \frac{n\varphi(m)}{m} + O(2^{\omega(m)}).$$

Putting everything together, we arrive at the equation,

$$\sum_{\substack{1 \leq j \leq n \\ \gcd(j,m)=1}} 1 = n \frac{\varphi(m)}{m} + O(2^{\omega(m)}),$$

thereby completing the proof. \square

References

- Chen, L.; Jordan, S.; Liu, Y.K.; Moody, D.; Peralta, R.; Perlner, R.; Smith-Tone, D. *Report on Post-Quantum Cryptography*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2016. [\[CrossRef\]](#)
- Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; IEEE: Piscataway, NJ, USA, 1994; pp. 124–134. [\[CrossRef\]](#)
- Regev, O. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM* **2005**, *56*, 34:1–34:40. [\[CrossRef\]](#)
- National Institute of Standards and Technology (NIST). *Post-Quantum Cryptography*; NIST: Gaithersburg, MD, USA, 2017.
- Kuwakado, H.; Morii, M. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In Proceedings of the 2010 IEEE International Symposium on Information Theory, Austin, TX, USA, 13–18 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 2682–2685. [\[CrossRef\]](#)

6. Kuwakado, H.; Morii, M. Security on the quantum-type Even-Mansour cipher. In Proceedings of the 2012 International Symposium on Information Theory and its Applications, Honolulu, HI, USA, 28–31 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 312–316.
7. Kaplan, M.; Leurent, G.; Leverrier, A.; Naya-Plasencia, M. Breaking symmetric cryptosystems using quantum period finding. In *Advances in Cryptology—CRYPTO 2016*; Robshaw, M., Katz, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 207–237. [[CrossRef](#)]
8. Santoli, T.; Schaffner, C. Using Simon’s algorithm to attack symmetric-key cryptographic primitives. *Quantum Information & Computation*; Rinton Press: Paramus, NJ, USA, 2017; pp. 65–78. [[CrossRef](#)]
9. Simon, D.R. On the power of quantum computation. *SIAM J. Comput.* **1997**, *26*, 1474–1483. [[CrossRef](#)]
10. Gagliardoni, T.; Hülsing, A.; Schaffner, C. Semantic security and indistinguishability in the quantum world. *Advances in Cryptology—CRYPTO 2016*; Robshaw, M., Katz, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 60–89. [[CrossRef](#)]
11. Bleichenbacher, D. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. *Advances in Cryptology—CRYPTO ’98*; Krawczyk, H., Ed.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 1–12. [[CrossRef](#)]
12. Boneh, D.; Zhandry, M. Secure signatures and chosen ciphertext security in a quantum computing world. *Advances in Cryptology—CRYPTO 2013*; Canetti, R., Garay, J.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 361–379. [[CrossRef](#)]
13. Boneh, D.; Zhandry, M. Quantum-secure message authentication codes. *Advances in Cryptology—EUROCRYPT 2013*; Johansson, T., Nguyen, P.Q., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 592–608. [[CrossRef](#)]
14. Broadbent, A.; Jeffery, S. Quantum homomorphic encryption for circuits of low T-gate complexity. *Advances in Cryptology—CRYPTO 2015*; Gennaro, R., Robshaw, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 609–629. [[CrossRef](#)]
15. Zhandry, M. How to construct quantum random functions. In Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science, New Brunswick, NJ, USA, 20–23 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 679–687. [[CrossRef](#)]
16. Zhandry, M. A note on quantum-secure PRPs. *arXiv* **2016**, arXiv:1611.05564.
17. Bernstein, E.; Vazirani, U. Quantum complexity theory. *SIAM J. Comput.* **1997**, *26*, 1411–1473. [[CrossRef](#)]
18. Alkim, E.; Bos, J.W.; Ducas, L.; Longa, P.; Mironov, I.; Naehrig, M.; Nikolaenko, V.; Peikert, C.; Raghunathan, A.; Stebila, D.; et al. *FrodoKEM—Learning With Errors Key Encapsulation*; Submission to the NIST Post-Quantum Cryptography standardization project, Round 2. Revised July 2, 2019. unpublished work.
19. Lindner, R.; Peikert, C. Better key sizes (and attacks) for LWE-based encryption. In *Topics in Cryptology—CT-RSA 2011*; Kiayias, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 319–339. [[CrossRef](#)]
20. Lyubashevsky, V.; Peikert, C.; Regev, O. A toolkit for ring-LWE cryptography. In *Advances in Cryptology—EUROCRYPT 2013*; Johansson, T., Nguyen, P.Q., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 35–54. [[CrossRef](#)]
21. Lyubashevsky, V.; Peikert, C.; Regev, O. On ideal lattices and learning with errors over rings. *J. ACM* **2013**, *60*, 43:1–43:35. [[CrossRef](#)]
22. Grilo, A.B.; Kerenidis, I.; Zijlstra, T. Learning with errors is easy with quantum samples. *arXiv* **2017**, arXiv:1702.08255.
23. Fujisaki, E.; Okamoto, T. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology—CRYPTO 1999*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 537–554.
24. Goldreich, O.; Levin, L.A. A hard-core predicate for all one-way functions. In Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing (STOC ’89), Seattle, WA, USA, 14–17 May 1989; ACM: New York, NY, USA, 1989; pp. 25–32. [[CrossRef](#)]
25. Goldreich, O.; Goldwasser, S.; Micali, S. How to construct random functions. *J. ACM* **1986**, *33*, 792–807. [[CrossRef](#)]
26. Nayak, A. Optimal lower bounds for quantum automata and random access codes. In Proceedings of the 40th Annual Symposium on Foundations of Computer Science, New York City, NY, USA, 17–19 October 1999; pp. 369–376. [[CrossRef](#)]
27. Ambainis, A.; Leung, D.; Mancinska, L.; Ozols, M. Quantum random access codes with shared randomness. *arXiv*, **2008**, arXiv:0810.2937.
28. Kitaev, A.Y. Quantum measurements and the abelian stabilizer problem. *arXiv* **1995**, arXiv:quant-ph/9511026.

29. Hales, L.; Hallgren, S. An improved quantum Fourier transform algorithm and applications. In Proceedings of the 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, CA, USA, 12–14 November 2000; pp. 515–525. [[CrossRef](#)]
30. Goldreich, O. *Foundations of Cryptography: Volume 2, Basic Applications*; Cambridge University Press: Cambridge, UK, 2009.
31. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2001.
32. Alkim, E.; Ducas, L.; Pöppelmann, T.; Schwabe, P. Post-quantum key exchange—A new hope. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; USENIX Association: Austin, TX, USA, 2016; pp. 327–343.
33. Bshouty, N.H.; Jackson, J.C. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM J. Comput.* **1998**, *28*, 1136–1153. [[CrossRef](#)]
34. Tomczak-Jaegermann, N. The moduli of smoothness and convexity and the Rademacher averages of the trace classes S_p ($1 \leq p < \infty$). *Stud. Math.* **1974**, *50*, 163–182.
35. Humphries, P. Reference for Euler Totient Function Identity? Mathematics Stack Exchange. Available online: <https://math.stackexchange.com/q/2889027> (accessed on 8 May 2012).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).