Quantum CCA-Secure PKE, Revisited

Navid Alamati[*] Varun Maram[†]

Abstract

Security against chosen-ciphertext attacks (CCA) concerns privacy of messages even if the adversary has access to the decryption oracle. While the classical notion of CCA security seems to be strong enough to capture many attack scenarios, it falls short of preserving the privacy of messages in the presence of *quantum* decryption queries, i.e., when an adversary can query a superposition of ciphertexts.

Boneh and Zhandry (CRYPTO 2013) defined the notion of quantum CCA (qCCA) security to guarantee privacy of messages in the presence of *quantum* decryption queries. However, their construction is based on an exotic cryptographic primitive (namely, identity-based encryption with security against *quantum* queries), for which only one instantiation is known. In this work, we comprehensively study qCCA security for public-key encryption (PKE) based on both generic cryptographic primitives and concrete assumptions, yielding the following results:

- We show that key-dependent message secure encryption (along with PKE) is sufficient to realize qCCA-secure PKE. This yields the first construction of qCCA-secure PKE from the LPN assumption.
- We prove that hash proof systems imply qCCA-secure PKE, which results in the first instantiation of PKE with qCCA security from (isogeny-based) group actions.
- We extend the notion of adaptive TDFs (ATDFs) to the quantum setting by introducing *quantum* ATDFs, and we prove that quantum ATDFs are sufficient to realize qCCA-secure PKE. We also show how to instantiate quantum ATDFs from the LWE assumption.
- We show that a single-bit qCCA-secure PKE is sufficient to realize a multi-bit qCCA-secure PKE by extending the completeness of bit encryption for CCA security to the quantum setting.

[*] VISA Research.

[†] SandboxAQ. The work was done while the author was an intern at VISA Research (and a PhD student at ETH Zürich).

1 Introduction

Security against chosen-ciphertext attacks (CCA) concerns privacy of messages against adversaries whose power is beyond eavesdropping. In a CCA-secure public-key encryption (PKE) scheme, encryptions of two adverserially chosen messages are computationally indistinguishable even if the adversary has access to the decryption oracle [NY90, DDN91, RS92]. CCA security is considered to be the *de facto* notion of security for PKE [Sho98], and there has been a long line of works studying CCA security from various cryptographic assumptions (e.g., [CS02, PW08, RS09, KMP14, KW19, KMT19, HLLG19, HKW20, ADMP20]).

Quantum CCA security. While the classical notion of CCA security seems to be strong enough to capture many attack scenarios, it falls short of preserving the privacy of messages in the presence of *quantum* decryption queries, i.e., when an adversary can query a superposition of ciphertexts and receive a superposition of their decryptions. To capture such attack scenarios, Boneh and Zhandry [BZ13b] defined the notion of quantum CCA (IND-qCCA, or qCCA for short) security, and as they pointed out, issuing quantum decryption queries (and the notion of qCCA in general) capture the security of a natural model of ubiquitous quantum computing environment where users encrypt messages on a quantum computer.

Comparing qCCA to (classical) CCA security, Boneh and Zhandry showed that there exist PKE schemes that satisfy (post-quantum) CCA security but those schemes can be immediately broken under qCCA attacks, concluding that qCCA security is stronger than CCA security. In terms of instantiations, they demonstrated a construction of qCCA-secure PKE from an identity-based encryption (IBE) scheme with selective security against *quantum* (secret key) queries (by relying on the generic transformation of [BCHK07]). They showed how to realize qCCA security from the learning with errors (LWE) assumption by observing that the LWE-based IBE scheme of [ABB10] can be shown to satisfy selective security against quantum queries.

However, despite considerable progress in the area of quantum cryptography in recent years (e.g., [FKS$^+$13, BJSW16, RZ21, KNY21, BCKM21, MY22b, AQY22]), to the best of our knowledge, the aforementioned blueprint of Boneh and Zhandry remains the *only* way of realizing qCCA-secure PKE from either generic or concrete assumptions in the *standard* model after nearly a decade; it is worth pointing out that a line of recent works, namely [XY19, LW21, SGX23], do provide generic constructions of qCCA-secure PKE albeit in the *idealized* quantum random oracle model (QROM). This is in contrast with the (classical) CCA-secure PKE for which we have a variety of constructions from concrete (post-quantum) assumptions such as learning parity with noise (LPN) or variants of isogeny-based group actions (e.g., variants of CSIDH [CLM$^+$18]). Thus we ask the following natural question:

Can we construct qCCA-secure PKE from a wider class of concrete assumptions such as LPN or isogeny-based group actions?

Furthermore, in the past two decades, there has been remarkable progress on realizing CCA security (in a black-box manner) from generic assumptions starting from hash proof systems [CS02] and lossy/correlated-secure/adaptive trapdoor functions (TDFs) [PW08, RS09, KMO10] to more recent ones based on circular security and injective trapdoor functions [KW19, KMT19, HKW20]. On the other hand, the only generic cryptographic primitive which is known to imply qCCA-secure PKE is IBE with security against *quantum* queries. Therefore, even in terms of generic cryptographic assumptions, qCCA security is much less understood compared to its classical counterpart. This is despite the fact that for many other cryptographic primitives (e.g., symmetric-key primitives, digital signatures, passively secure PKE, etc.), the gap between classical and quantum security is little to none [Zha12, BZ13a, BZ13b]. In particular, in case of symmetric-key encryption (SKE), qCCA security has already been shown to be implied by (post-quantum) CCA-secure SKE (or by the minimal assumption of post-quantum one-way functions) [BZ13b]. This leads to the following question:

Can we build qCCA-secure PKE from the same set of cryptographic primitives (or a subset thereof) that imply CCA-secure PKE?

On a related note, it has long been known that bit encryption is complete for CCA security [Ms09]. Specifically, given a *single-bit* CCA-secure PKE one can construct a *multi-bit* PKE with CCA security. However, such an implication is not known for quantum CCA security. So we ask the following pertinent question:

Is bit encryption complete for quantum CCA-secure PKE?

Quantum security for adaptive TDFs. In 2010, Kiltz *et al.* [KMO10] introduced the notion of *adaptive* trapdoor functions, which can be viewed as a "deterministic" form of CCA-secure PKE. Informally, a trapdoor function is said to be adaptive if it remains one way even if the adversary is given access to an inversion oracle. The authors of [KMO10] demonstrated a construction of CCA-secure PKE from adaptive TDFs. They also showed how adaptive TDFs can be constructed from lossy or correlated-secure TDFs. This motivates us to ask the following question:

Does there exist a quantum analog of adaptive TDFs?

1.1 Our Contributions

We answer the questions described above in the affirmative by presenting the following results, narrowing the gap between CCA and qCCA security for PKE. In particular, our results for qCCA essentially match what is known for CCA security in terms of generic cryptographic primitives, while also yielding new instantiations of qCCA security from a variety of concrete assumptions. We refer to Figure 1 for a simplified overview of our results.

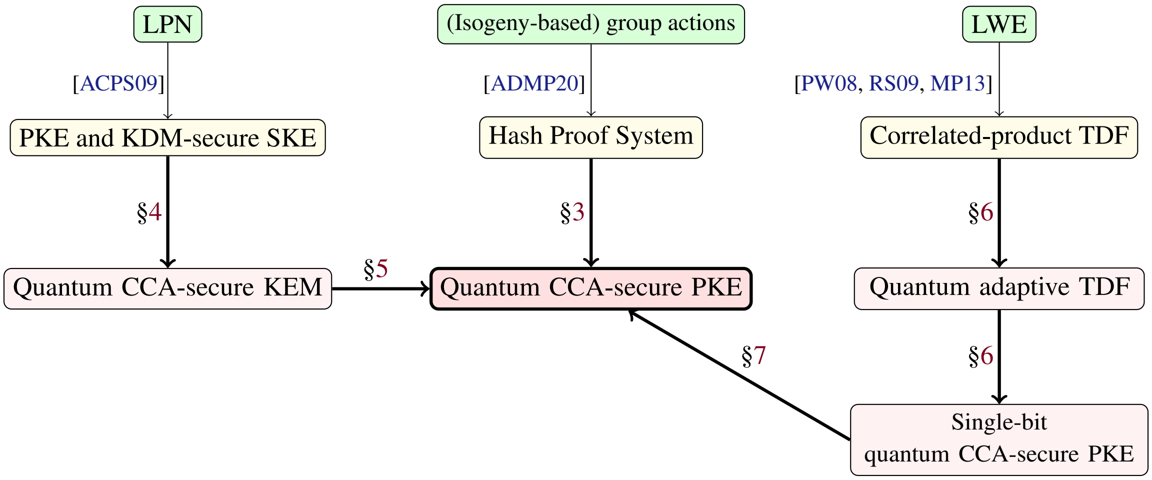

Figure 1: An overview of our results (our contributions are denoted by bold arrows)

Quantum CCA security from KDM-secure SKE and PKE. We show that the CCA-secure PKE construction of Kitagawa *et al.* [KMT19] from PKE and a key-dependent message (KDM) secure SKE also satisfies *quantum* CCA security if the underlying primitives satisfy post-quantum security. By plugging in the KDM-secure SKE construction of [ACPS09] from LPN, our result yields the first construction of qCCA-secure PKE from the LPN assumption. Along the way, we also prove that the KEM-DEM[1] hybrid encryption of [CS03] results in a qCCA-secure PKE if (1) the underlying KEM is qCCA-secure, and (2) the underlying DEM offers (post-quantum) one-time authenticated encryption security with respect to *classical* queries only.

Quantum CCA security from hash proof systems. We prove that the CCA-secure PKE construction of Cramer and Shoup [CS02] from hash proof systems also satisfies *quantum* CCA security if the underlying hash proof system satisfies post-quantum security. Coupled with the hash proof system construction of [ADMP20] from variants of CSIDH, our result yields the first construction of qCCA-secure PKE from isogeny-based group actions.

[1] Key Encapsulation Mechanism and Data Encapsulation Mechanism, respectively.

Completeness of bit encryption for qCCA security. We show the quantum analog of completeness of bit encryption for CCA security by showing that single-bit qCCA-secure PKE is sufficient to realize multi-bit PKE with qCCA security. Our result extends the framework of [HLW12] to the quantum setting without any additional assumption.

Quantum adaptive TDFs. We extend the notion of adaptive TDFs (ATDFs) to the quantum setting by introducing the notion of *quantum* ATDFs. We also extend the result of Kiltz *et al.* [KMO10] to the quantum setting by showing that quantum ATDFs are sufficient to realize qCCA-secure PKE. In addition, in terms of constructions, we describe how to build quantum adaptive TDFs from (post-quantum) correlated-product TDFs, which in turn yields an instantiation from the LWE assumption [PW08, RS09, MP13].

Remark 1.1. We emphasize that all the above constructions are *classical PKE schemes* in the sense that they can be implemented on classical computers; this is in contrast to so-called *quantum PKE schemes* constructed in recent works such as [MY22a, Col23, BGH$^+$23] which inherently require quantum machinery, e.g., requiring public keys to be quantum states. We essentially share the same goal as [BZ13b] to construct classical cryptosystems which remain secure when eventually implemented on quantum computers wherein adversaries potentially get quantum decryption access to such devices.

Remark 1.2. Quantum CCA security restricts the adversary to *classical* challenge messages (but still allowing *quantum* decryption queries; see Definition 2.2 in Section 2). Recently, Chevalier *et al.* [CEV22] introduced a new quantum security notion that captures indistinguishability under quantum chosen-ciphertext attacks (i.e., qIND-qCCA security), where the indistinguishability of ciphertexts holds even for quantum superpositions of messages. They also show that qIND-qCCA secure PKE can be realized from any qCCA-secure PKE. Hence, by plugging in their lifting theorem in our results, we get new constructions of qIND-qCCA secure PKE from a variety of generic cryptographic primitives as well as concrete assumptions.

1.2 Technical Overview

We provide a simplified technical overview of our results. For the ease of exposition, we focus on a particular construction as a warm-up example. We begin by recalling the construction of CCA-secure PKE scheme from *correlated-product* trapdoor functions (CP-TDFs) in [RS09], and next we prove its *quantum* CCA security (while relying on the post-quantum security of the underlying CP-TDF). Looking ahead, even though we show in Section 6 that CP-TDFs imply the stronger notion of quantum adaptive TDFs (which in turn are sufficient to realize qCCA security), for this overview we aim to highlight the main aspects of extending the classical CCA security proofs of [RS09] to the quantum setting. It is worth pointing out that the following analysis can be extended in a straightforward fashion to prove qCCA security of the LWE-based PKE construction in [BZ13b] that relies on *quantum* selective-secure IBE.

Informally speaking, a CP-TDF is a family of trapdoor functions $\{f_{\mathsf{ek}}\}_{\mathsf{ek}\in\mathcal{K}}$ such that the following family $\{f_{\mathsf{ek}_1,\ldots,\mathsf{ek}_t}\}_{(\mathsf{ek}_1,\ldots,\mathsf{ek}_t)\in\mathcal{K}^t}$ defined by

$$f_{\mathsf{ek}_1,\ldots,\mathsf{ek}_t}(x) = (f_{\mathsf{ek}_1}(x), \ldots, f_{\mathsf{ek}_t}(x))$$

is also one-way, i.e., one-wayness is guaranteed even if one uses the same input (but independently chosen evaluation keys). The PKE construction of [RS09] from CP-TDFs proceeds as follows. The public key consists of t pairs of (random) functions $(f_{\mathsf{ek}_1^0}, f_{\mathsf{ek}_1^1}), \ldots, (f_{\mathsf{ek}_t^0}, f_{\mathsf{ek}_t^1})$, where each ek_i^b is sampled from $\mathcal{K}$, and the secret key consists of the trapdoors $(\mathsf{td}_1^0, \mathsf{td}_1^1), \ldots, (\mathsf{td}_t^0, \mathsf{td}_t^1)$, where each td_i^b is a trapdoor corresponding to $f_{\mathsf{ek}_i^b}$. To encrypt a bit m, first generate $(\mathsf{vk}, \mathsf{sk})$ for a one-time signature $(\mathsf{Sign}, \mathsf{Ver})$ such that $\mathsf{vk} = (\mathsf{vk}_1, \ldots, \mathsf{vk}_t) \in \{0,1\}^t$, then choose a random input x and compute the following:

$$\forall i \in [t] : y_i = f_{\mathsf{ek}_i^{\mathsf{vk}_i}}(x), \quad \mathsf{ct}_1 = \mathsf{m} \oplus h(x), \quad \mathsf{ct}_2 \leftarrow \mathsf{Sign}(\mathsf{sk}, (y_1, \ldots, y_t, \mathsf{ct}_1)),$$

where h is a hard-core predicate of $f_{\mathsf{ek}_1,\ldots,\mathsf{ek}_t}$, and output $(\mathsf{vk}, y_1, \ldots, y_t, \mathsf{ct}_1, \mathsf{ct}_2)$. To decrypt such a ciphertext, check $\mathsf{Ver}(\mathsf{vk}, ((y_1, \ldots, y_t, \mathsf{ct}_1)), \mathsf{ct}_2) = 0$. If so, return $\perp$. Otherwise, for every $i \in [t]$, invert y_i using $\mathsf{td}_i^{\mathsf{vk}_i}$ to obtain x_i. If $x_1 = x_2 = \cdots = x_k$, output $h(x_1) \oplus \mathsf{ct}_1$; otherwise, output $\perp$.

To sketch the CCA security proof of the above PKE construction in [RS09], consider the following distinguisher $\mathcal{D}$ where $\mathcal{D}$ gets t functions $f_{\mathsf{ek}_1}, \ldots, f_{\mathsf{ek}_t}$ along with t values $y_1^* = f_{\mathsf{ek}_1}(x^*), \ldots, y_t^* = f_{\mathsf{ek}_t}(x^*)$ for a uniformly random x^*, and a challenge bit b which is either $h(x^*)$ or a random bit. $\mathcal{D}$ can simulate the CCA security game for $\mathcal{A}$ by first generating $(\mathsf{vk}^*, \mathsf{sk}^*)$ for one-time signature, and computing the public key $(f_{\mathsf{ek}_1^0}, f_{\mathsf{ek}_1^1}), \ldots, (f_{\mathsf{ek}_t^0}, f_{\mathsf{ek}_t^1})$ to be sent to $\mathcal{A}$ as follows: first, $\mathcal{D}$ sets

$$\forall i \in [t] : f_{\mathsf{ek}_i^{\mathsf{vk}_i^*}} := f_{\mathsf{ek}_i},$$

and then for the remaining part of the public key, $\mathcal{D}$ samples $\mathsf{ek}_i^{1-\mathsf{vk}_i^*}$ along with the corresponding $\mathsf{td}_i^{1-\mathsf{vk}_i^*}$. Now observe that $\mathcal{D}$ can answer any of $\mathcal{A}$'s decryption queries by using the trapdoor $\mathsf{td}_i^{1-\mathsf{vk}_i^*}$ for some index i such that $\mathsf{vk}_i \neq \mathsf{vk}_i^*$. The challenge ciphertext is later computed as $\mathsf{ct}^* = (\mathsf{vk}^*, y_1^*, \ldots, y_t^*, \mathsf{ct}_1^*, \mathsf{ct}_2^*)$ where $\mathsf{ct}_1^* = \mathsf{m} \oplus b$ for a random message $\mathsf{m} \in \{0,1\}$ and $\mathsf{ct}_2^* \leftarrow \mathsf{Sign}(\mathsf{sk}^*, (y_1^*, \ldots, y_t^*, \mathsf{ct}_1^*))$. As before, $\mathcal{D}$ can respond to the rest of $\mathcal{A}$'s decryption queries while responding $\perp$ when the query is equal to ct^*. If $\mathcal{A}$ later guesses the message m correctly, $\mathcal{D}$ outputs that $b = h(x^*)$; otherwise, $\mathcal{D}$ outputs that b is a random bit.

Observe that $\mathcal{D}$ perfectly simulates the decryption algorithm above *only* when $\mathsf{vk} \neq \mathsf{vk}^*$ since otherwise $\mathcal{D}$ does not have access to the corresponding trapdoor for decryption. However, as argued in [RS09], the probability that $\mathcal{A}$ makes a query $\overline{\mathsf{ct}} = (\mathsf{vk}^*, y_1, \ldots, y_t, \mathsf{ct}_1, \mathsf{ct}_2)$ with $\mathsf{Ver}(\mathsf{vk}^*, ((y_1, \ldots, y_t, \mathsf{ct}_1)), \mathsf{ct}_2) = 1$ is negligible thanks to the unforgeability of $(\mathsf{Sign}, \mathsf{Ver})$. So if $\mathcal{A}$ makes such a query $\overline{\mathsf{ct}}$, then a signature forger simulating the CCA security game towards $\mathcal{A}$ can use $\overline{\mathsf{ct}}$ to break the unforgeability of $(\mathsf{Sign}, \mathsf{Ver})$.

Coming to the qCCA setting however (see Definition 2.2 in Section 2 for a formal definition of qCCA security for PKE), where $\mathcal{A}$ can ask for the decryption of a quantum superposition of different ciphertexts, it is quite possible that ciphertexts $\overline{\mathsf{ct}}$ of the above form that induce a signature forgery are among the superposition. Therefore, if we want to repeat the *same* reduction above, it is not clear how a signature forger can "extract" $\overline{\mathsf{ct}}$-based forgeries from $\mathcal{A}$'s quantum queries. This brings us to the main tool we employ in our qCCA security proofs in this paper: the (generalized) One-Way To Hiding (OW2H) lemma [AHU19]. Informally, the lemma states that given two oracles $G, H : \mathcal{X} \to \mathcal{Y}$ whose outputs differ with respect to a set of inputs S, and an algorithm A that has *quantum* access to either G or H, the probability that A can distinguish between G and H is essentially bounded by the square root of the probability when measuring a random quantum oracle query made by A results in a classical state in S.

To resolve the issue above in the quantum setting, we use the OW2H lemma as follows. Let G be the original decryption oracle that $\mathcal{A}$ has quantum access to in the qCCA security game. We modify G to obtain a *new* quantum decryption oracle H which rejects (i.e., returns $\perp$) ciphertexts $\overline{\mathsf{ct}}$ of the above form; the difference set S described above precisely includes such ciphertexts $\overline{\mathsf{ct}}$. Now note that $\mathcal{D}$ can simulate the modified decryption oracle H towards $\mathcal{A}$ *even in the quantum setting* using the same CCA simulation strategy described above while simply rejecting ciphertexts $\overline{\mathsf{ct}}$ that contain vk^* as above. In our qCCA security proof, we argue that $\mathcal{A}$'s winning probability in the original qCCA security game (where it has access to G) changes by at most a negligible amount when it has access to H instead. Here we invoke the OW2H lemma to show the quantum indistinguishability of oracles G and H by bounding the probability when measuring a random decryption query made by $\mathcal{A}$ to H results in a ciphertext $\overline{\mathsf{ct}} \in S$ that induces a signature forgery. This follows straightforwardly from the *classical* unforgeability of the one-time signature scheme. Namely, in the corresponding reduction, a signature forger simulates quantum access to H towards $\mathcal{A}$ and randomly measures one of $\mathcal{A}$'s decryption queries; if the measurement results in $\overline{\mathsf{ct}}$, then the forger succeeds. We remark that the reason we only require *classical* security from the signature is that, the forger queries a *classical* message to its one-time signing oracle (as opposed to querying messages in superposition, which is accounted for by *quantum* security of signatures in [BZ13b]) to obtain a signature $\mathsf{ct}_2^* \leftarrow \mathsf{Sign}(\mathsf{sk}^*, (y_1^*, \ldots, y_t^*, \mathsf{ct}_1^*))$, when computing the challenge ciphertext.

This is a common theme across most of our qCCA security proofs, i.e., we extend the classical CCA security proofs of PKE constructions considered in this paper to the quantum CCA setting by first identifying the modifications made to the decryption oracle in the classical CCA analysis and then arguing the quantum indistinguishability between these oracles in our qCCA analysis by relying on the generalized OW2H lemma.[1] However, it is not always easy to bound the probability of "measuring a decryption query to S" when applying the OW2H lemma in our qCCA analysis, in contrast to our proof sketch above where we relied on the unforgeability of a one-time signature. For instance, as we will see later, we show that single-bit qCCA secure PKE implies multi-bit qCCA secure PKE by extending the framework

[1] Sometimes such changes are made *implicitly* to the decryption oracle, as was the case in the CCA security proof of [RS09] sketched above; but we have to make these modifications more explicit in our qCCA proofs in order to apply the OW2H lemma.

of [HLW12] to the quantum setting, and bounding the measurement probability requires another "nested" application of the OW2H lemma, which introduces further subtleties in the proof.

On a related note, the original version of the OW2H lemma (introduced in [Unr14]) handled only *random* oracles G and H, and found widespread use in proving security of cryptosystems in the QROM [BDF+11]. Later, [AHU19] introduced a generalized version of the OW2H lemma which not only allowed G and H to have an arbitrary output distribution but also allowed the distinguisher's auxiliary input to be arbitrarily correlated with G, H, and the difference set S. This allows us to apply the generalized OW2H lemma in our qCCA security proofs with respect to quantum decryption oracles, which are *not* synonymous with random oracles. To the best of our knowledge, our results include the *first* application of the OW2H lemma in the context of proving quantum CCA security of PKE schemes in the *standard* model. This is in contrast to relying on, arguably, more complicated techniques such as the *compressed oracle* framework introduced in [Zha19], which was used to analyze qCCA security of PKE schemes obtained from the *Fujisaki-Okamoto transformation* [FO13] (in the QROM). In fact, the (q)CCA security proof in [Zha19] was later found to have some subtle gaps in it [DFMS22]. Furthermore, the work of [Unr20] showed a framework for *formally* verifying post-quantum security proofs of cryptosystems that involve applications of the OW2H lemma; this can be seen as evidence of the relative simplicity of the OW2H proof technique.

Remark 1.3. We remark that we chose to present the toy example above (for qCCA security) for the sake of brevity and providing intuition, since otherwise explaining our main results (e.g., based on hash proof systems or KDM security) would require recalling a rather lengthy preliminary background. We refer to Sections 3−7 for our detailed results.

Remark 1.4. We note that one might use other well-known techniques (for example, the Gentle Measurement Lemma; see [BBC+21, Theorem 1], an adaptation of [BBBV97, Theorem 3.3]) to show the indistinguishability of the quantum decryption oracles G and H in the overview above. We chose to use the OW2H lemma because of its relative simplicity and that it does not result in a significant loss in security proofs, as evidenced by the widespread usage of this technique in analyzing CCA security of some proposals (in the QROM) in the NIST standardization process for post-quantum cryptography. Hence, one of the technical contributions of this paper is finding a novel application for the OW2H lemma (a popular QROM proof technique) in the *standard* model, in order to establish the quantum CCA security of various standard model PKE constructions.

2 Preliminaries

Notations. The value of $(x \overset{?}{=} y)$ is defined to be 1 if $x = y$ and 0 otherwise. For a positive integer n, we denote $[n]$ to be the set $\{1, 2, \ldots, n\}$. We use $\lambda \in \mathbb{N}$ to denote the security parameter. For a finite set S, we write $x \leftarrow S$ to denote that x is uniformly at random sampled from S, unless stated otherwise. $x \parallel y$ denotes their concatenation. $|x|$ denotes bit-length of the encoding of x. For probabilistic algorithms we use $y \leftarrow \mathcal{A}(x)$ to denote a (randomized) output of $\mathcal{A}$ on input x; we also sometimes specify the randomness r used in $\mathcal{A}$ as $y \leftarrow \mathcal{A}(x; r)$. We omit writing λ when it is clear from context.

Quantum(-accessible) Oracles. We refer the reader to [NC00] for the basics of quantum computation and information. Here we recall a basic and useful fact about quantum computation.

> **Fact:** *Any classical computation can also be implemented on a quantum computer, and also any function that has an efficient classical algorithm can be implemented efficiently as a quantum-accessible oracle.*

Given an algorithm $\mathcal{A}$ and a (classical) function $O : \{0,1\}^m \to \{0,1\}^n$, we use $\mathcal{A}^{|O\rangle}$ to denote that $\mathcal{A}$ has *quantum* access to an oracle implementing O. To be more precise, $\mathcal{A}^{|O\rangle}$ can make standard superposition queries $\sum_{x,z} \psi_{x,z} |x, z\rangle$ to the quantum oracle $|O\rangle$ and get $\sum_{x,z} \psi_{x,z} |x, z \oplus O(x)\rangle$ as a response; here x and z are arbitrary m-bit and n-bit strings respectively. (We omit the "ket" notation in $\mathcal{A}^{|O\rangle}$, and instead just write $\mathcal{A}^{O}$ when it is clear from context.)

Lemma 2.1 (Generalized OW2H [AHU19, Theorem 3]). *Let $\mathcal{S} \subseteq \mathcal{X}$ be a random subset. Let $G, H : \mathcal{X} \to \mathcal{Y}$ be random oracles satisfying $G(x) = H(x)$ for every $x \notin \mathcal{S}$. Let z be a random bit string. ($\mathcal{S}, G, H, z$ may have arbitrary joint distribution.) Let A be a quantum oracle algorithm making at most q quantum queries to its corresponding oracle* *(either G or H). Let B^H be an oracle algorithm that on input z does the following: picks $i \leftarrow \{1, \ldots, q\}$, runs $A^H(z)$ until (just before) the i-th query, measures all query input registers in the computational basis, and outputs the set $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$ of measurement outcomes (if A makes less than i queries, the measurement outcomes are taken to be $\perp \notin \mathcal{X}$). Let,*

$$\begin{aligned}
P_{\text{left}} &= \Pr[1 \leftarrow A^H(z)] \\
P_{\text{right}} &= \Pr[1 \leftarrow A^G(z)] \\
P_{\text{guess}} &= \Pr[\mathcal{S} \cap \mathcal{T} \neq \emptyset : \mathcal{T} \leftarrow B^H(z)]
\end{aligned}$$

Then, $|P_{\text{left}} - P_{\text{right}}| \leq 2q\sqrt{P_{\text{guess}}}$. The same result also holds with B^G instead of B^H in the definition of P_{guess}.

Public-Key Encryption (PKE). A PKE scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ is a triple of algorithms, where the algorithm Gen on input 1^λ generates a pair of public and secret keys $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$; the probabilistic algorithm Enc encrypts a message $\mathsf{m} \in \mathcal{M}$ as $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ and the deterministic Dec decrypts a ciphertext ct as $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ (or returns $\perp$).

For a function $\gamma : \mathbb{N} \mapsto [0, 1]$, we say that PKE is γ-spread if for every key pair $(\mathsf{pk}, \mathsf{sk})$, message $\mathsf{m} \in \mathcal{M}$, and ciphertext ct, we have

$$\Pr_{r \leftarrow \mathcal{R}}[\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r)] \leq 2^{-\gamma}$$

where $\mathcal{R}$ is the set of all possible randomness that can be sampled in Enc. In particular, we say that PKE is *well-spread* if $2^{-\gamma}$ is negligible in λ.

Let $\varepsilon : \mathbb{N} \mapsto [0, 1]$ be a function. We say that $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is ε-almost-all-keys correct if we have

$$\Pr_{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)}[\exists(\mathsf{m}, \mathsf{r}) \text{ s.t. } \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; \mathsf{r}) \neq \mathsf{m})] \leq \varepsilon(\lambda).$$

We also call key pairs $(\mathsf{pk}, \mathsf{sk})$ under which the above decryption error occurs as "erroneous."

qCCA-Secure PKE. Below we define quantum CCA security for PKE, which is an extension of classical CCA security to the quantum setting.

Definition 2.2. A PKE scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be *qCCA* secure if for every QPT adversary $\mathcal{A}$, the following quantity $\mathbf{Adv}^{\text{qCCA}}_{\mathsf{PKE},\mathcal{A}}$ is negligible:

$$\mathbf{Adv}^{\text{qCCA}}_{\mathsf{PKE},\mathcal{A}} = \left| \Pr\left[b = b' : \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda); b \leftarrow \{0, 1\} \\ (\mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \leftarrow \mathcal{A}^{|O_\perp(\mathsf{sk},\cdot)\rangle}(\mathsf{pk}) \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b); b' \leftarrow \mathcal{A}^{|O_{\mathsf{ct}^*}(\mathsf{sk},\cdot)\rangle}(\mathsf{ct}^*, \mathsf{st}) \end{array} \right] - \frac{1}{2} \right|$$

where the function $O_{\tilde{\mathsf{ct}}}(\mathsf{sk}, \cdot)$ is defined as

$$O_{\tilde{\mathsf{ct}}}(\mathsf{sk}, \mathsf{ct}) = \begin{cases} \perp & \text{if } \mathsf{ct} = \tilde{\mathsf{ct}}, \\ \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) & \text{otherwise,} \end{cases}$$

where st denotes some arbitrary state information and ct^* is computed in the *challenge* phase. We require the messages m_0 and m_1 to be of the same length. $\mathcal{A}$ has access to $|O_\perp(\mathsf{sk}, \cdot)\rangle$ and $|O_{\mathsf{ct}^*}(\mathsf{sk}, \cdot)\rangle$ in the *pre-challenge* phase and *post-challenge* phase, respectively. As in the qCCA security definition for encryption in [BZ13b], we also encode $\perp$ to be a bit-string outside the message space $\mathcal{M}$ in order to properly define the result $z \oplus \perp$ in the output register of $|O_{\tilde{\mathsf{ct}}}(\mathsf{sk}, \cdot)\rangle$ described above.

The notion of CCA security for PKE schemes differs from Definition 2.2 in that the adversary $\mathcal{A}$ has *classical* access to $O_\perp(\mathsf{sk}, \cdot)$ and $O_{\mathsf{ct}^*}(\mathsf{sk}, \cdot)$.

Injective Trapdoor Functions. A trapdoor function (TDF) $\mathsf{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ is a triple of algorithms satisfying the following one-wayness property, where the algorithm Gen on input 1^λ generates a pair of evaluation and trapdoor keys $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$, $\mathsf{Eval}(\mathsf{ek}, \cdot)$ implements a function $f_{\mathsf{ek}}(\cdot)$ over $\{0,1\}^\lambda$ and $\mathsf{Invert}(\mathsf{td}, \cdot)$ implements its inverse function $f_{\mathsf{ek}}^{-1}(\cdot)$ (along with outputting $\perp$ for invalid inputs). Note that we require TDFs to be injective.

Definition 2.3. A trapdoor function $\mathsf{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ satisfies (post-quantum) one-wayness if for every QPT inverter $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathrm{OW}}_{\mathsf{TDF},\mathcal{A}} = \Pr\left[x = x' : \begin{array}{c} (\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda); x \leftarrow \{0,1\}^\lambda \\ y^* \leftarrow \mathsf{Eval}(\mathsf{ek}, x); x' \leftarrow \mathcal{A}(\mathsf{ek}, y^*) \end{array}\right] \leq \mathrm{negl}\,.$$

Definition 2.4. We say that the function $\mathrm{GL} : \{0,1\}^\lambda \to \{0,1\}$ is a hardcore predicate for a (post-quantum) TDF $\mathsf{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ if for every QPT distinguisher $\mathcal{A}$, the following quantity $\mathbf{Adv}^{\mathrm{hcDist}}_{\mathrm{GL},\mathcal{D}}$ is negligible:

$$\mathbf{Adv}^{\mathrm{hcDist}}_{\mathrm{GL},\mathcal{D}} = \left|\Pr\left[b = b' : \begin{array}{c} (\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda); b \leftarrow \{0,1\} \\ x \leftarrow \{0,1\}^\lambda; y^* \leftarrow \mathsf{Eval}(\mathsf{ek}, x); h_0 = \mathrm{GL}(x) \\ h_1 \leftarrow \{0,1\}; b' \leftarrow \mathcal{A}(\mathsf{ek}, y^*, h_b) \end{array}\right] - \frac{1}{2}\right|.$$

The work of [GL89] showed that there *must* exist a hardcore predicate for *any* one-way (trapdoor) function.

qCCA-Secure KEM. A key-encapsulation mechanism $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ with session-key[1] space $\mathcal{K}$ is a triple of algorithms, where the algorithm Gen on input 1^λ generates a pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$; the algorithm Encaps generates a session key and the corresponding ciphertext as $(\mathsf{ct}, \mathsf{k}) \leftarrow \mathsf{Encaps}(\mathsf{pk})$ and the deterministic Decaps returns a session key (or an error $\perp$) from a ciphertext as $\mathsf{k} \leftarrow \mathsf{Decaps}(\mathsf{sk}, \mathsf{ct})$.

Let $\varepsilon : \mathbb{N} \mapsto [0,1]$ be a function. We say that KEM is ε-almost-all-keys correct if we have

$$\Pr_{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)}[\exists \mathsf{r} \text{ s.t. } \mathsf{Encaps}(\mathsf{pk}; \mathsf{r}) = (\mathsf{ct}, \mathsf{k}) \wedge \mathsf{Decaps}(\mathsf{sk}, \mathsf{ct}) \neq \mathsf{k}] \leq \varepsilon(\lambda).$$

We call pairs $(\mathsf{pk}, \mathsf{sk})$ under which the decapsulation error occurs as "erroneous."

Definition 2.5. A KEM $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ is said to be *qCCA* secure if for every QPT adversary $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathrm{qCCA}}_{\mathsf{KEM},\mathcal{A}} = \left|\Pr\left[b = b' : \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda); b \leftarrow \{0,1\} \\ (\mathsf{ct}^*, \mathsf{k}_1^*) \leftarrow \mathsf{Encaps}(\mathsf{pk}), \mathsf{k}_0^* \leftarrow \mathcal{K} \\ b' \leftarrow \mathcal{A}^{|O_{\mathsf{ct}^*}(\mathsf{sk},\cdot)\rangle}(\mathsf{pk}, (\mathsf{ct}^*, \mathsf{k}_b^*)) \end{array}\right] - \frac{1}{2}\right| \leq \mathrm{negl},$$

where the function $O_{\tilde{\mathsf{ct}}}(\mathsf{sk}, \cdot)$ is defined as

$$O_{\tilde{\mathsf{ct}}}(\mathsf{sk}, \mathsf{ct}) = \begin{cases} \perp & \text{if } \mathsf{ct} = \tilde{\mathsf{ct}}, \\ \mathsf{Decaps}(\mathsf{sk}, \mathsf{ct}) & \text{otherwise.} \end{cases}$$

(We also encode $\perp$ to be a bitstring outside $\mathcal{K}$ to properly define the value $z \oplus \perp$ in the output register of $|O_{\tilde{\mathsf{ct}}}(\mathsf{sk}, \cdot)\rangle$ described above.)

The notion of CCA security for KEMs differs from Definition 2.5 in that the adversary $\mathcal{A}$ has *classical* access to the oracle $O_{\mathsf{ct}^*}(\mathsf{sk}, \cdot)$.

Coming to the CPA security of KEMs, it is convenient to define the security notion for the following *multi-challenge* experiment; this multi-challenge version of CPA security is polynomially equivalent to the single-challenge version via a standard hybrid argument. A KEM $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ is said to be CPA secure if for any polynomial $n = n(\lambda)$ and every QPT adversary $\mathcal{A}$, we have that the following quantity $\mathbf{Adv}^{\mathrm{mCPA}}_{\mathsf{KEM},n,\mathcal{A}}$ is negligible:

[1] Also sometimes referred to as "encapsulated key" in this paper.

$$\mathbf{Adv}^{\text{mCPA}}_{\mathsf{KEM},n,\mathcal{A}} = \left| \Pr\left[b = b' : \begin{array}{c} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda); b \leftarrow \{0,1\} \\ \forall i \in [n]\ (\mathsf{ct}^*_i, \mathsf{k}^*_{i,1}) \leftarrow \mathsf{Encaps}(\mathsf{pk}), \mathsf{k}^*_{i,0} \leftarrow \mathcal{K} \\ b' \leftarrow \mathcal{A}(\mathsf{pk}, (\mathsf{ct}^*_i, \mathsf{k}^*_{i,b})_{i\in[n]}) \end{array} \right] - \frac{1}{2} \right|.$$

Authenticated Encryption Scheme. A secret-key encryption (SKE) scheme $\mathsf{SKE} = (\mathsf{K},\mathsf{E},\mathsf{D})$ is a triple of algorithms where the algorithm K on input 1^λ generates a key $\mathsf{k} \leftarrow \mathsf{K}(1^\lambda)$; the probabilistic algorithm E encrypts a message m as $\mathsf{ct} \leftarrow \mathsf{E}(\mathsf{k},\mathsf{m})$ and the deterministic (decryption) D decrypts a ciphertext ct as $\mathsf{m} \leftarrow \mathsf{D}(\mathsf{k},\mathsf{ct})$ (or returns an error $\perp$). We also assume perfect correctness of our SKE schemes.

Definition 2.6. An SKE scheme $\mathsf{SKE} = (\mathsf{K},\mathsf{E},\mathsf{D})$ is a *one-time* authenticated encryption scheme if the following two properties hold:

- CPA security: For every QPT adversary $\mathcal{A}$, it holds

$$\mathbf{Adv}^{\text{CPA}}_{\mathsf{SKE},\mathcal{A}} = \left| \Pr\left[b = b' : \begin{array}{l} \mathsf{k} \leftarrow \mathsf{K}(1^\lambda); b \leftarrow \{0,1\} \\ (\mathsf{m}_0,\mathsf{m}_1,\mathsf{st}) \leftarrow \mathcal{A}(1^\lambda) \\ \mathsf{ct}^* \leftarrow \mathsf{E}(\mathsf{k},\mathsf{m}_b); b' \leftarrow \mathcal{A}(\mathsf{ct}^*,\mathsf{st}) \end{array} \right] - \frac{1}{2} \right| \leq \text{negl},$$

  where m_0 and m_1 are of the same length and st is state information.

- Ciphertext integrity (INT-CTXT security): For every QPT adversary $\mathcal{B}$, we have

$$\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{SKE},\mathcal{B}} = \Pr\left[\mathsf{win} = 1 : \begin{array}{l} \mathsf{k} \leftarrow \mathsf{K}(1^\lambda); \mathsf{win} = 0 \\ (\mathsf{m},\mathsf{st}) \leftarrow \mathcal{B}^{O_\perp(\mathsf{k},\cdot)}(1^\lambda) \\ \mathsf{ct}^* \leftarrow \mathsf{E}(\mathsf{k},\mathsf{m}); \mathcal{B}^{O_{\mathsf{ct}^*}(\mathsf{k},\cdot)}(\mathsf{ct}^*,\mathsf{st}) \end{array} \right] \leq \text{negl},$$

  where the flag "win" is set to 1 if $\mathcal{B}$ makes a query to any of the oracles $O_\perp(\mathsf{k},\cdot)$ or $O_{\mathsf{ct}^*}(\mathsf{k},\cdot)$ such that the corresponding response is *not* $\perp$. Here the oracle $O_{\tilde{\mathsf{ct}}}(\mathsf{k},\cdot)$ is defined as

$$O_{\tilde{\mathsf{ct}}}(\mathsf{k},\mathsf{ct}) = \begin{cases} \perp & \text{if } \mathsf{ct} = \tilde{\mathsf{ct}}, \\ \mathsf{D}(\mathsf{k},\mathsf{ct}) & \text{otherwise.} \end{cases}$$

  Note that $\mathcal{B}$ only has *classical* access to the oracles.

One-time security stems from the fact that $\mathcal{A}$ and $\mathcal{B}$ have *one-time* access to the encryption oracle $\mathsf{E}(\mathsf{k},\cdot)$.

Key-Dependent Message (KDM) Security. A function is said to be a *projection function* if each of its output bits depends on at most a single bit of its input. Let $\mathsf{SKE} = (\mathsf{K},\mathsf{E},\mathsf{D})$ be an SKE with key space $\mathcal{K}$ and message space $\mathcal{M}$. Let $\mathcal{P}$ be the family of projection functions with domain $\mathcal{K}$ and range $\mathcal{M}$.

Definition 2.7. An SKE scheme $\mathsf{SKE} = (\mathsf{K},\mathsf{E},\mathsf{D})$ is said to be *one-time* KDM secure with respect to projection functions if for every QPT adversary $\mathcal{A}$, we have

$$\mathbf{Adv}^{\text{KDM}}_{\mathsf{SKE},\mathcal{P},\mathcal{A}} = \left| \Pr\left[b = b' : \begin{array}{l} \mathsf{k} \leftarrow \mathsf{K}(1^\lambda); b \leftarrow \{0,1\} \\ (f_0,f_1,\mathsf{st}) \leftarrow \mathcal{A}(1^\lambda) \\ \mathsf{ct}^* \leftarrow \mathsf{E}(\mathsf{k},f_b(\mathsf{k})); b' \leftarrow \mathcal{A}(\mathsf{ct}^*,\mathsf{st}) \end{array} \right] - \frac{1}{2} \right| \leq \text{negl}$$

where $f_0, f_1 \in \mathcal{P}$ and st is some arbitrary state information.

Target Collision-Resistant Hash Functions. A keyed hash function $\mathsf{Hash} = (\mathsf{HGen}, \mathsf{H})$ is a pair of algorithms where the algorithm HGen on input 1^λ generates a hash key $\mathsf{hk} \leftarrow \mathsf{HGen}(1^\lambda)$; the deterministic (evaluation) algorithm H on input a value $x \in \{0,1\}^*$ outputs a hash value $y \in \{0,1\}^\lambda$, i.e., $y = \mathsf{H}(\mathsf{hk}, x)$.

Definition 2.8. A hash function $\mathsf{Hash} = (\mathsf{HGen}, \mathsf{H})$ is *target collision-resistant* if for every QPT adversary $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathsf{TCR}}_{\mathsf{Hash},\mathcal{A}} = \Pr\left[\begin{matrix}\mathsf{H}(\mathsf{hk}, x') = \mathsf{H}(\mathsf{hk}, x) \\ \wedge\, x' \neq x\end{matrix} : \begin{matrix}(x, \mathsf{st}) \leftarrow \mathcal{A}(1^\lambda) \\ \mathsf{hk} \leftarrow \mathsf{HGen}(1^\lambda); x' \leftarrow \mathcal{A}(\mathsf{hk}, \mathsf{st})\end{matrix}\right] \leq \mathrm{negl},$$

where st is some arbitrary state information.

Note that target collision-resistant hash functions can be constructed from any one-way function [Rom90].

3 Quantum CCA Security from Hash Proof Systems

Cramer and Shoup [CS02] introduced the notion of *hash proof systems*, which provides a generic framework to construct CCA-secure PKE. In this section, we show how one can also obtain *quantum* CCA-secure PKE from hash proof systems while relying on the *same* statistical (i.e., universality and smoothness) and computational[1] properties that were used for building CCA-secure PKE. By plugging in the hash proof system construction of [ADMP20] from isogeny-based group actions (e.g., variants of CSIDH), we obtain the first realization of qCCA-secure PKE from isogeny-based assumptions.

We begin by recalling the definition of universal hash proof systems (also known as *projective* hash functions) as in [CS02, ADMP20].

Definition 3.1. Let $\Lambda : K \times \Sigma \to \Gamma$ be an efficiently computable function, and let $L \subset \Sigma$. Also, let $\alpha : K \to S$ be a projection function. We say that the tuple $\Pi = (\Lambda, K, S, \Sigma, \Gamma, L)$ is a universal hash proof system if the following holds:

- *Samplability:* There exist efficient algorithms to sample from Σ and from K. In addition, there exists an efficient algorithm to sample from L along with a witness w that proves membership in L.
- *Subset Membership Problem:* If $\sigma_0 \leftarrow L$ and $\sigma_1 \leftarrow \Sigma \setminus L$, it holds that σ_0 is computationally indistinguishable from σ_1, i.e., for any PPT distinguisher $\mathcal{D}$, the following is negligible:

$$\mathbf{Adv}^{\mathsf{SMP}}_{(\Sigma,L),\mathcal{D}} = \left|\Pr\left[b' = b : b \leftarrow \{0,1\}; \sigma_0 \leftarrow L, \sigma_1 \leftarrow \Sigma \setminus L, b' \leftarrow \mathcal{D}(\sigma_b)\right] - \frac{1}{2}\right|.$$

- *Projective Evaluation:* There exists an efficient algorithm $\mathsf{ProjEval}$ such that for any $k \in K$ and any $\sigma \in L$ with membership witness w, we have

$$\mathsf{ProjEval}(\alpha(k), \sigma, w) = \Lambda(k, \sigma).$$

- *Universality:* Π is said to be ε-universal if for any $\sigma \in \Sigma \setminus L$, $s \in S$, and $\gamma \in \Gamma$, we have

$$\Pr_{k \leftarrow K}[\Lambda(k, \sigma) = \gamma \mid \alpha(k) = s] \leq \varepsilon.$$

Universality$_2$ and smoothness. We recall two stronger notions for universal hash proof systems, namely universality$_2$ and smoothness, as in [CS02].

- *Universality$_2$:* A hash proof system $\Pi = (\Lambda, K, S, \Sigma, \Gamma, L)$ is ε-universal$_2$ if for any $\sigma, \sigma^* \in \Sigma$ such that $\sigma \in \Sigma \setminus (L \cup \{\sigma^*\})$, for any $s \in S$ and $\gamma, \gamma^* \in \Gamma$, we have

$$\Pr_{k \leftarrow K}[\Lambda(k, \sigma) = \gamma \mid \alpha(k) = s \wedge \Lambda(k, \sigma^*) = \gamma^*] \leq \varepsilon.$$

[1] The only difference is that we require the computational properties to hold in the presence of QPT adversaries (i.e., post-quantum security).

- *Smoothness:* A hash proof system $\Pi = (\Lambda, K, S, \Sigma, \Gamma, L)$ is ε-smooth if for any $\sigma \in \Sigma \setminus L$, $k \leftarrow K$ and $\gamma \leftarrow \Gamma$, the statistical distance between $(\alpha(k), \sigma, \Lambda(k, \sigma))$ and $(\alpha(k), \sigma, \gamma)$ is at most ε.

As in [CS02], we also define an *extended* hash proof system with a tuple of the form $\Pi = (\Lambda, K, S, \Sigma \times E, \Gamma, L \times E)$ associated with a finite set E (where E is going to be used for encoding messages). The only difference between an extended and an "ordinary" hash proof system is that to compute $\Lambda(k, \sigma, e)$ for $\sigma \in L$ and $e \in E$, the $\mathsf{ProjEval}$ algorithm takes as input $\alpha(k) \in S$, $\sigma \in L$, $e \in E$ and a witness w, i.e., $\mathsf{ProjEval}(\alpha(k), \sigma, e, w) = \Lambda(k, \sigma, e)$.

Construction. We recall the construction of CCA-secure PKE from universal (and smooth) hash proof systems [CS02]. We then proceed to show that the same construction also results in qCCA-secure PKE, assuming the post-quantum security of the underlying hash proof system.

Let $\Pi = (\Lambda, K, S, \Sigma, \Gamma, L)$ be an ε'-*smooth* hash proof system, and also let $\alpha : K \to S$ be its projection function. Let $\hat{\Pi} = (\hat{\Lambda}, \hat{K}, \hat{S}, \Sigma \times \Gamma, \hat{\Gamma}, L \times \Gamma)$ be an extended hash proof system with ε-*universality*$_2$, with $\hat{\alpha} : \hat{K} \to \hat{S}$ being the corresponding projection function. Consider the scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space Γ based on Π and $\hat{\Pi}$ as follows (note that we require Γ to be an abelian group wherein elements can be efficiently added and subtracted):

$\mathsf{Gen}(1^\lambda)$: Sample $k \leftarrow K$, $\hat{k} \leftarrow \hat{K}$, and compute $s = \alpha(k)$, $\hat{s} = \hat{\alpha}(\hat{k})$. Output

$$\mathsf{pk} = (s, \hat{s}), \quad \mathsf{sk} = (k, \hat{k}).$$

$\mathsf{Enc}(\mathsf{pk} = (s, \hat{s}), \mathsf{m})$: Sample $\sigma \in L$ with its witness w. Output $(\sigma, e, \hat{\gamma})$ where

$$\gamma = \mathsf{ProjEval}(s, \sigma, w), \quad e = \mathsf{m} + \gamma \in \Gamma, \quad \hat{\gamma} = \widehat{\mathsf{ProjEval}}(\hat{s}, \sigma, e, w).$$

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct} = (\sigma, e, \hat{\gamma}))$: Compute $\bar{\gamma} = \hat{\Lambda}(\hat{k}, \sigma, e)$. If $\hat{\gamma} = \bar{\gamma}$ do the following, else output $\perp$: compute $\gamma = \Lambda(k, \sigma) \in \Gamma$ and output $\mathsf{m} = e - \gamma \in \Gamma$.

Theorem 3.2. *If Π is a ε'-*smooth *hash proof system and $\hat{\Pi}$ is a ε-*universal$_2$ *extended hash proof system for negligible ε' and ε, then* PKE *is qCCA secure.*

At a high level, the proof proceeds similarly to that of [CS02] but the main difference is that in [CS02], the original Dec oracle is replaced with an alternative oracle, and then the ε-universality$_2$ property of $\hat{\Pi}$ is invoked in order to argue the (classical) indistinguishability of decryption oracles. In our case, we argue the *quantum* indistinguishability of these decryption oracles by relying on the generalized OW2H lemma [AHU19] (also referred to as Lemma 2.1 in the rest of this paper, as formally defined in Section 2), in addition to the *statistical* ε-universality$_2$ property of $\hat{\Pi}$.

Proof. Let $\mathcal{A}$ be any QPT adversary that breaks the qCCA security of PKE (see Definition 2.2) while making q quantum decryption queries, with q_{pre}/q_{post} decryption queries in the pre/post-challenge phase. Consider the following games.

Game 1: This is essentially the same as the qCCA game except for some minor changes.[1]

- Sample $k \leftarrow K$ and $\hat{k} \leftarrow \hat{K}$ and compute $s = \alpha(k)$, $\hat{s} = \hat{\alpha}(\hat{k})$. Set $\mathsf{pk} = (s, \hat{s})$ and $\mathsf{sk} = (k, \hat{k})$. Generate $\sigma^* \in L$ along with a corresponding witness w^*, and sample a random bit $b \leftarrow \{0, 1\}$.
- Forward pk to $\mathcal{A}$ and respond to $\mathcal{A}$'s quantum decryption queries using the description of $\mathsf{Dec}(\mathsf{sk}, \cdot)$ above, i.e., given a ciphertext $|\mathsf{ct}\rangle = |\sigma, e, \hat{\gamma}\rangle$ in the computational basis, compute $\bar{\gamma} = \hat{\Lambda}(\hat{k}, \sigma, e)$ and if $\hat{\gamma} = \bar{\gamma}$ then compute $\gamma = \Lambda(k, \sigma)$ and output $\mathsf{m} = e - \gamma$. Otherwise, output $\perp$.
- After receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, output $\mathsf{ct}^* = (\sigma^*, e^*, \hat{\gamma}^*)$ where

$$\gamma^* = \mathsf{ProjEval}(s, \sigma^*, w^*), \quad e^* = \mathsf{m}_b + \gamma^*, \quad \hat{\gamma}^* = \widehat{\mathsf{ProjEval}}(\hat{s}, \sigma^*, e^*, w^*).$$

[1] Specifically, (σ^*, w^*) is generated in the *pre-challenge phase* (which is going to be used in the challenge phase). However, this change does not affect $\mathcal{A}$'s view.

- Respond to $\mathcal{A}$'s quantum decryption queries in the normal way as above, but this time making sure to reject ciphertexts that are equal to ct^*.
- $\mathcal{A}$ terminates with an output $b' \in \{0,1\}$.

Game 2: In this game, we modify the way $\mathsf{ct}^* = (\sigma^*, e^*, \hat{\gamma}^*)$ is computed. Specifically, instead of using pk to encrypt m_b, we use $\mathsf{sk} = (k, \hat{k})$ as follows:

$$\gamma^* = \Lambda(k, \sigma^*), \quad e^* = \mathsf{m}_b + \gamma^*, \quad \hat{\gamma}^* = \hat{\Lambda}(\hat{k}, \sigma^*, e^*).$$

Game 3: In this game, we sample σ^* uniformly from $\Sigma \setminus L$, instead of L. Note that we do not need a corresponding witness w^* as we are not using the $\mathsf{ProjEval}$ function (which requires a witness as input) anymore to encrypt m_b.

Game 4a: In this game, we modify the decryption oracle in the *pre-challenge* phase as follows. In addition to rejecting a ciphertext $(\sigma, e, \hat{\gamma})$ if $\hat{\Lambda}(\hat{k}, \sigma, e) \neq \hat{\gamma}$, the modified oracle also rejects the ciphertext if $\sigma \notin L$.

Game 4b: Here we modify the decryption oracle in the *post-challenge* phase as follows. In addition to rejecting a ciphertext $(\sigma, e, \hat{\gamma})$ if $\hat{\Lambda}(\hat{k}, \sigma, e) \neq \hat{\gamma}$, the modified oracle also rejects the ciphertext if $\sigma \notin L$.[1]

Game 5: In this game, we modify the challenge phase as follows. Instead of computing γ^* as $\gamma^* = \Lambda(k, \sigma^*)$, we sample γ^* uniformly from Γ.

We define $W^{(j)}$, for $j \in \{1, 2, 3, 4a, 4b, 5\}$, to be the event that $\mathcal{A}$ succeeds in guessing the bit b (i.e., $b' = b$) in Game j. By definition, we have

$$\mathbf{Adv}^{\mathsf{qCCA}}_{\mathsf{PKE},\mathcal{A}} = \left| \Pr[W^{(1)}] - \frac{1}{2} \right|.$$

We now have the following in the subsequent games.

Lemma 3.3. $\Pr[W^{(1)}] = \Pr[W^{(2)}]$.

Proof. Note that since $\sigma^* \in L$ (where w^* is the corresponding witness), by the projective evaluation property we have $\gamma^* = \mathsf{ProjEval}(s, \sigma^*, w^*) = \Lambda(k, \sigma^*)$ and $\hat{\gamma}^* = \widehat{\mathsf{ProjEval}}(\hat{s}, \sigma^*, e^*, w^*) = \hat{\Lambda}(\hat{k}, \sigma^*, e^*)$. □

Lemma 3.4. *There exists a distinguisher $\mathcal{D}$ that solves the subset membership problem of Π (and $\hat{\Pi}$) such that $|\Pr[W^{(2)}] - \Pr[W^{(3)}]| = 2 \cdot \mathbf{Adv}^{\mathsf{SMP}}_{(\Sigma,L),\mathcal{D}}$.*

Proof. The description of $\mathcal{D}$ is as follows: on input $\sigma^* \in \Sigma$, the distinguisher $\mathcal{D}$ samples $\mathsf{sk} = (k, \hat{k}) \leftarrow K \times \hat{K}$ and forwards the corresponding public key $\mathsf{pk} = (\alpha(k), \hat{\alpha}(\hat{k}))$ to the qCCA adversary $\mathcal{A}$. It then samples $b \leftarrow \{0,1\}$ and answers $\mathcal{A}$'s quantum decryption queries as in Game 2 using sk. When $\mathcal{A}$ provides $(\mathsf{m}_0, \mathsf{m}_1)$, the distinguisher $\mathcal{D}$ computes the challenge ciphertext as follows: it computes $\gamma^* = \Lambda(k, \sigma^*)$ using sk. Next, $\mathcal{D}$ computes $e^* = \mathsf{m}_b + \gamma^*$ and $\hat{\gamma}^* = \hat{\Lambda}(\hat{k}, \sigma^*, e^*)$, and forwards $(\sigma^*, e^*, \hat{\gamma}^*)$ to $\mathcal{A}$. The distinguisher $\mathcal{D}$ proceeds to respond to $\mathcal{A}$'s quantum decryption queries again as in Game 2 using sk while making sure to reject ciphertexts equal to ct^*. Finally, when $\mathcal{A}$ terminates with a bit b', the distinguisher $\mathcal{D}$ outputs 1 if $b = b'$ and outputs 0 otherwise. It is easy to see that $\mathcal{D}$ perfectly simulates Game 2 (respectively, Game 3) if $\sigma^* \in L$ (respectively, $\sigma^* \in \Sigma \setminus L$). Therefore, we have

$$|\Pr[W^{(2)}] - \Pr[W^{(3)}]| = 2 \cdot \mathbf{Adv}^{\mathsf{SMP}}_{(\Sigma,L),\mathcal{D}}.$$ □

Lemma 3.5. $|\Pr[W^{(3)}] - \Pr[W^{(4a)}]| \leq 2q_{pre}\sqrt{\varepsilon}$.

Proof. Here we use Lemma 2.1 to bound $|\Pr[W^{(3)}] - \Pr[W^{(4a)}]|$. In the context of applying Lemma 2.1, let A be a quantum oracle algorithm which receives as input the hash proof systems Π and $\hat{\Pi}$ along with the random values $k \leftarrow K$ and $\hat{k} \leftarrow \hat{K}$ (namely, the secret key $\mathsf{sk} = (k, \hat{k})$); i.e., $z = ((\Pi, \hat{\Pi}), (k, \hat{k}))$. A also has quantum access either to the original decryption oracle $G := \mathsf{Dec}(\mathsf{sk}, \cdot)$ used in Game 3 or to the modified decryption oracle H used in Game 4a. Note that the outputs of oracles G and H differ with respect to the set of ciphertexts $S = \{\mathsf{ct} = (\sigma, e, \hat{\gamma}) | \sigma \notin$

[1] As usual, ciphertexts equal to the challenge ciphertext ct^* will also be rejected.

$L \wedge \hat{\Lambda}(\hat{k}, \sigma, e) = \hat{\gamma}\}$. Next, A proceeds to perfectly simulate either Game 3 or Game 4a towards the qCCA adversary $\mathcal{A}$ (depending on whether it has access to G or H) as follows: A first samples $b \leftarrow \{0,1\}$ and forwards the public key $(s, \hat{s}) = (\alpha(k), \hat{\alpha}(\hat{k}))$ to $\mathcal{A}$. Then A *strictly* uses its quantum oracle (i.e., G or H) to answer $\mathcal{A}$'s queries in the pre-challenge phase, i.e., A does not use the secret key $(k, \hat{k})$ *directly* for decryption here. To compute the challenge ciphertext $\mathsf{ct}^* = (\sigma^*, e^*, \hat{\gamma}^*)$ in the challenge phase, A uses the private keys k and $\hat{k}$ for encrypting m_b just as in Game 3 (and 4a) above. Finally, A proceeds to answer the rest of $\mathcal{A}$'s quantum decryption queries in the post-challenge phase as in Game 3, this time using the secret key $(k, \hat{k})$ (and *not* the oracles G or H), while at the same time rejecting ciphertexts that are equal to ct^*. Finally A outputs 1 if and only if $\mathcal{A}$ outputs $b' = b$.

Observe that in the context of Lemma 2.1, $\Pr[W^{(3)}] = \Pr[1 \leftarrow A^G(z)]$ and $\Pr[W^{(4a)}] = \Pr[1 \leftarrow A^H(z)]$. Thus, we have $|\Pr[W^{(3)}] - \Pr[W^{(4a)}]| \leq 2q_{pre}\sqrt{P_{\text{guess}}}$ where P_{guess} is the probability of the event when measurement of a random quantum decryption query made by $\mathcal{A}$ in the pre-challenge phase of Game 4a results in a ciphertext $(\sigma, e, \hat{\gamma}) \in S$.

In order to bound the probability P_{guess} in Game 4a, we first condition on fixed hash proof systems Π and $\hat{\Pi}$, as well as fixed values of k, $\hat{s}$, and $\mathcal{A}$'s random coins. These values completely determine the public key received by $\mathcal{A}$ in Game 4a, the quantum decryption queries made by $\mathcal{A}$ in the *pre-challenge phase*, the corresponding responses of the decryption oracle (note that in Game 4a, the decryption oracle rejects ciphertexts $(\sigma, e, \hat{\gamma})$ when $\sigma \notin L$; hence, when $\sigma \in L$, we only need the keys k and $\hat{s}$ for decryption), and the values m_0 and m_1 chosen by $\mathcal{A}$ in the challenge phase. Now consider any ciphertext $\mathsf{ct} = (\sigma, e, \hat{\gamma})$ which is a result of measuring any quantum decryption query made by $\mathcal{A}$ in the pre-challenge phase. If E is an event in this conditional probability space, then we denote the associated probability of the event in this space as $\Pr_{\text{cond}}[E]$. In the next step, we want to bound the following quantity:

$$\Pr_{\text{cond}}[(\sigma, e, \hat{\gamma}) \in S] = \Pr_{\text{cond}}[\sigma \notin L \wedge \hat{\Lambda}(\hat{k}, \sigma, e) = \hat{\gamma}] \leq \Pr_{\text{cond}}[\hat{\Lambda}(\hat{k}, \sigma, e) = \hat{\gamma} \mid \sigma \notin L].$$

In this conditional probability space where σ, e, and $\hat{\gamma}$ are fixed, along with the other values fixed above, note that $\hat{k}$ is still uniformly distributed over $\hat{K}$ conditioned on $\hat{\alpha}(\hat{k}) = \hat{s}$. Hence, from the ε-universality$_2$ property of $\hat{\Pi}$, we have $\Pr_{\text{cond}}[\hat{\Lambda}(\hat{k}, \sigma, e) = \hat{\gamma} \mid \sigma \notin L] \leq \varepsilon$. Thus, from a simple averaging argument over this conditional probability space, it follows that $P_{\text{guess}} \leq \varepsilon$ in Game 4a. Therefore, it follows that

$$|\Pr[W^{(3)}] - \Pr[W^{(4a)}]| \leq 2q_{pre}\sqrt{P_{\text{guess}}} \leq 2q_{pre}\sqrt{\varepsilon}. \qquad \square$$

Lemma 3.6. $|\Pr[W^{(4a)}] - \Pr[W^{(4b)}]| \leq 2q_{post}\sqrt{\varepsilon}$.

Proof. We again use Lemma 2.1 as above to bound $|\Pr[W^{(4a)}] - \Pr[W^{(4b)}]|$. We first simulate the pre-challenge phase of Game 4a (and 4b) as follows: let A_{pre} be an algorithm which receives as input the hash proof systems Π and $\hat{\Pi}$ along with a randomly chosen $\mathsf{sk} = (k, \hat{k})$ generated as in Game 4a above. A_{pre} samples $b \leftarrow \{0,1\}$ and forwards $\mathsf{pk} = (s, \hat{s}) = (\alpha(k), \hat{\alpha}(\hat{k}))$ to $\mathcal{A}$. It responds to $\mathcal{A}$'s quantum decryption queries in the pre-challenge phase as in Game 4a using sk: for any $\mathsf{ct} = (\sigma, e, \hat{\gamma})$, it returns $\perp$ whenever $\sigma \notin L$ (note that A_{pre} need not be efficient); otherwise, it returns $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$. After receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, the algorithm A_{pre} computes $\mathsf{ct}^* = (\sigma^*, e^*, \hat{\gamma}^*)$ using sk as in Game 4a (and 4b) above. It then forwards ct^* to $\mathcal{A}$. In addition, in the context of applying Lemma 2.1, A_{pre} forwards the input $z = ((\Pi, \hat{\Pi}), (\mathsf{pk}, \mathsf{sk}), \mathsf{ct}^*, b)$ to a quantum oracle algorithm A_{post}. The algorithm A_{post} also has quantum access either to the corresponding post-challenge decryption oracle $G = \mathsf{Dec}(\mathsf{sk}, \cdot)$ in Game 4a (which also rejects ciphertexts equal to ct^*) or to the modified post-challenge decryption oracle H in Game 4b. Note that the outputs of G and H differ with respect to the set $S = \{\mathsf{ct} = (\sigma, e, \hat{\gamma}) | \mathsf{ct} \neq \mathsf{ct}^* \wedge \sigma \notin L \wedge \hat{\Lambda}(\hat{k}, \sigma, e) = \hat{\gamma}\}$. Next, A_{post} proceeds to simulate the post-challenge phase of Game 4a or Game 4b towards $\mathcal{A}$ (depending on whether it has access to G or H) by forwarding $\mathcal{A}$'s quantum decryption queries to its own oracle (i.e., G or H) and returning the corresponding output. Finally, A_{post} outputs 1 if and only if $\mathcal{A}$ outputs $b' = b$.

Observe that $\Pr[W^{(4a)}] = \Pr[1 \leftarrow A^G_{post}(z)]$ and $\Pr[W^{(4b)}] = \Pr[1 \leftarrow A^H_{post}(z)]$. By applying Lemma 2.1 we have $|\Pr[W^{(4a)}] - \Pr[W^{(4b)}]| \leq 2q_{post}\sqrt{P_{\text{guess}}}$ where P_{guess} is essentially the probability of the event when measurement of a random quantum decryption query made by $\mathcal{A}$ in the post-challenge phase of Game 4b results in a non-challenge ciphertext $\mathsf{ct} = (\sigma, e, \hat{\gamma}) \in S$.

In order to bound the probability P_{guess} in Game 4b, we condition on fixed hash proof systems Π and $\hat{\Pi}$, fixed values of k, $\hat{s}$, and $\mathcal{A}$'s random coins as in our analysis of the pre-challenge phase in the previous lemma. Moreover, we also condition on the fixed values of b and σ^* in the challenge phase (which determine γ^* and e^*) as well as a fixed value of $\hat{\gamma}^* = \hat{\Lambda}(\hat{k}, \sigma^*, e^*)$. These values completely determine all of the quantum decryption queries made by $\mathcal{A}$ in the *post-challenge phase* and the corresponding responses of the decryption oracle in Game 4b. Now consider any ciphertext $\mathsf{ct} = (\sigma, e, \hat{\gamma})$ which is a result of the measurement of any quantum decryption query made by $\mathcal{A}$ in the post-challenge phase. Using the same notation of $\Pr_{\text{cond}}[\cdot]$ as in the proof of the previous lemma to denote conditional probabilities, we want to bound:

$$\begin{aligned}\Pr_{\text{cond}}[(\sigma, e, \hat{\gamma}) \in S] &= \Pr_{\text{cond}}[(\sigma, e, \hat{\gamma}) \neq (\sigma^*, e^*, \hat{\gamma}^*) \wedge \sigma \notin L \wedge \hat{\Lambda}(\hat{k}, \sigma, e) = \hat{\gamma}] \\ &\leq \Pr_{\text{cond}}[\hat{\Lambda}(\hat{k}, \sigma, e) = \hat{\gamma} \mid (\sigma, e, \hat{\gamma}) \neq (\sigma^*, e^*, \hat{\gamma}^*) \wedge \sigma \notin L].\end{aligned}$$

If $(\sigma, e) = (\sigma^*, e^*)$, then since $(\sigma, e, \hat{\gamma}) \neq (\sigma^*, e^*, \hat{\gamma}^*)$ we have $\hat{\Lambda}(\hat{k}, \sigma, e) \neq \hat{\gamma}$ with certainty; that is, $\Pr_{\text{cond}}[\hat{\Lambda}(\hat{k}, \sigma, e) = \hat{\gamma} \mid (\sigma, e, \hat{\gamma}) \neq (\sigma^*, e^*, \hat{\gamma}^*) \wedge \sigma \notin L] = 0$.

On the other hand, if $(\sigma, e) \neq (\sigma^*, e^*)$, then in this conditional probability space where σ, e, and $\hat{\gamma}$ are fixed, along with the other values fixed above, note that $\hat{k}$ is uniformly distributed conditioned on $\hat{\alpha}(\hat{k}) = \hat{s}$ *and* $\hat{\Lambda}(\hat{k}, \sigma^*, e^*) = \hat{\gamma}^*$. Thus, by the ε-universality$_2$ property of $\hat{\Pi}$, we have

$$\Pr_{\text{cond}}[\hat{\Lambda}(\hat{k}, \sigma, e) = \hat{\gamma} \mid (\sigma, e, \hat{\gamma}) \neq (\sigma^*, e^*, \hat{\gamma}^*) \wedge \sigma \notin L] \leq \varepsilon.$$

By a simple averaging argument, it follows that $P_{\text{guess}} \leq \varepsilon$, which implies

$$|\Pr[W^{(4a)}] - \Pr[W^{(4b)}]| \leq 2q_{post}\sqrt{P_{\text{guess}}} \leq 2q_{post}\sqrt{\varepsilon}. \qquad \square$$

Remark 3.7. Note that in our application of the OW2H lemma above (and also in the rest of our qCCA security proofs below), we first make an explicit distinction between the "pre-challenge" and "post-challenge" decryption oracles and then apply the OW2H lemma *separately* to the respective oracles. The reason is that the OW2H lemma, in its current form, is not directly applicable to *stateful* oracles (which is the case for decryption oracles in the qCCA security game). Nevertheless, we believe that a one-shot application of a "stateful"-version of the OW2H lemma would not lead to improved security bounds compared to our two-fold application of the plain OW2H lemma.

Lemma 3.8. $|\Pr[W^{(4b)}] - \Pr[W^{(5)}]| \leq \varepsilon'$.

Proof. We construct a (potentially inefficient) distinguisher $\mathcal{D}'$ as follows: on input the hash proof systems Π, $\hat{\Pi}$ and a tuple $(\alpha(k), \sigma^*, \gamma^*)$ where $\sigma^* \leftarrow \Sigma \setminus L$, $k \leftarrow K$, and an element $\gamma^* \in \Gamma$ (where $\mathcal{D}'$ is supposed to determine whether $\gamma^* = \Lambda(k, \sigma^*)$ or $\gamma^* \leftarrow \Gamma$), the distinguisher $\mathcal{D}'$ first samples $\hat{k} \leftarrow \hat{K}$ and forwards $\mathsf{pk} = (\alpha(k), \hat{\alpha}(\hat{k}))$ to $\mathcal{A}$. It then samples $b \leftarrow \{0, 1\}$ and proceeds to answer $\mathcal{A}$'s queries as in Game 4b (and 5) as follows: for any $\mathsf{ct} = (\sigma, e, \hat{\gamma})$, return $\perp$ whenever $\sigma \notin L$. Otherwise, compute $\bar{\gamma} = \hat{\Lambda}(\hat{k}, \sigma, e)$ and check if $\bar{\gamma} = \hat{\gamma}$. If not, output $\perp$. Otherwise, find a witness w corresponding to $\sigma \in L$ and compute $\gamma = \mathsf{ProjEval}(\alpha(k), \sigma, w)$, and return $\mathsf{m} = e - \gamma$.[1] In the challenge phase, when $\mathcal{A}$ provides $(\mathsf{m}_0, \mathsf{m}_1)$, the distinguisher $\mathcal{D}'$ computes $\mathsf{ct}^* = (\sigma^*, e^*, \hat{\gamma}^*)$ as follows: it computes $e^* = \mathsf{m}_b + \gamma^*$ and $\hat{\gamma}^* = \hat{\Lambda}(\hat{k}, \sigma^*, e^*)$ using $\hat{k}$. Next, $\mathcal{D}'$ proceeds to respond to the rest of $\mathcal{A}$'s queries as in the pre-challenge phase while rejecting ciphertexts equal to ct^*. Finally, when $\mathcal{A}$ terminates with a bit b', the distinguisher $\mathcal{D}'$ outputs 1 if $b = b'$ and outputs 0 otherwise. Observe that $\mathcal{D}'$ perfectly simulates Game 4b or Game 5 (depending on its input). Therefore, by the ε'-smoothness property of Π, it follows that the statistical distance between $(\alpha(k), \sigma^*, \gamma^* = \Lambda(k, \sigma^*))$ and $(\alpha(k), \sigma^*, \gamma^* \leftarrow \Gamma)$ is bounded by ε', as required. □

Lemma 3.9. $\Pr[W^{(5)}] = 1/2$.

Proof. The lemma follows by observing that the view of $\mathcal{A}$ is independent of b. □

[1] For a *fixed* $\sigma \in L$, finding a corresponding witness w may not be efficient. However, note that $\mathcal{D}'$ is potentially inefficient since our proof relies on a *statistical* property of hash proof systems, namely ε'-smoothness (Definition 3.1).

By putting together all of the above bounds, it follows that

$$\mathbf{Adv}^{\mathrm{qCCA}}_{\mathsf{PKE},\mathcal{A}} \leq 2 \cdot \mathbf{Adv}^{\mathrm{SMP}}_{(\Sigma,L),\mathcal{D}} + 2q\sqrt{\varepsilon} + \varepsilon',$$

which establishes the qCCA security of PKE. □

Quantum CCA-secure PKE from isogeny-based group actions. We remark that an ε'-smooth and ε-universal$_2$ hash proof system (where ε' and ε are negligible) can be generically constructed from a $1/2$-universal hash proof system by relying on the leftover hash lemma (as shown by [CS02]), and for the above construction one can take Γ to be the group of (fixed-length) bit strings with xor operation. Thus, all one needs to realize qCCA-secure PKE is a $1/2$-universal hash proof system. In particular, by relying the hash proof system construction of [ADMP20], Theorem 3.2 immediately yields a quantum CCA-secure PKE from isogeny-based group actions (e.g., variants of CSIDH). We note that the transformation of Cramer and Shoup [CS02] to construct a negligibly smooth and universal hash proof system from a $1/2$-universal hash proof system is also valid in a *quantum* setting since it neither requires a computational assumption nor does it involve interacting with an oracle. Namely, the transformation of [CS02] is entirely *statistical*, and only relies on simple statistical techniques/lemmas, e.g., parallelization and the leftover hash lemma (Section 3.5 of [CS02]).

4 Quantum CCA Security from PKE and KDM-Secure SKE

Kitagawa *et al.* [KMT19] showed how to realize CCA-secure KEM/PKE given any CPA-secure KEM/PKE and any SKE scheme with one-time KDM security (for projection functions). In this section, we prove that their construction also satisfies qCCA security while relying on (post-quantum security of) the same building blocks.

4.1 Quantum CCA-Secure KEM

We recall the KEM construction of [KMT19] from the following building blocks (see Section 2 for formal definitions of various security notions):

- a CPA-secure KEM $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$;
- a one-time KDM-secure SKE (for projection functions) $\mathsf{SKE} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$; and
- a target-collision resistant hash function $\mathsf{Hash} = (\mathsf{HGen}, \mathsf{H})$.

These building blocks are required to have the following properties:

- The session key space of KEM and the randomness space of Encaps are $\{0,1\}^{4\lambda}$ and $\{0,1\}^{\lambda}$, respectively. The secret key space of SKE is $\{0,1\}^{n}$ and plaintext space is $\{0,1\}^{n\cdot\lambda+\ell}$. The range of H is $\{0,1\}^{\lambda}$.
- The size of the range of $\mathsf{Decaps}(\mathsf{sk},\cdot)$ (excluding $\perp$) for any sk in the support of $\mathsf{Gen}(1^{\lambda})$ is at most 2^{λ}.

Consider the scheme $\overline{\mathsf{KEM}} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Encaps}}, \overline{\mathsf{Decaps}})$ described in Figure 2 and Figure 3. Correctness of $\overline{\mathsf{KEM}}$ can be verified as in [KMT19]. We proceed to prove the quantum CCA security of $\overline{\mathsf{KEM}}$ via the following theorem.

Theorem 4.1. *Let* KEM, SKE*, and* Hash *be as described above with security against QPT adversaries. If* KEM *is almost-all-keys correct*[1] *then* $\overline{\mathsf{KEM}}$ *is qCCA secure.*

Proof. Let $\mathcal{A}$ be any QPT adversary that breaks the qCCA security of $\overline{\mathsf{KEM}}$ while making q quantum decapsulation queries. Let KEM be ε-almost-all-keys correct. Our proof proceeds with a similar sequence of games as in [KMT19]. For the sake of completeness, we provide the descriptions of the games as follows.

Game 1: This is essentially identical to qCCA game for $\overline{\mathsf{KEM}}$ except for a few changes in the ordering.

[1] This correctness notion is analogous to the almost-all-keys correctness defined for PKE schemes in Section 2.

```
$\overline{\mathsf{Gen}}(1^\lambda)$:

  $(\mathsf{pk}^\mathsf{b}, \mathsf{sk}^\mathsf{b}) \leftarrow \mathsf{Gen}(1^\lambda)$    for $\mathsf{b} \in \{0,1\}$
  $\mathbf{a}_i, \mathbf{c} \leftarrow \{0,1\}^{4\lambda}$            for $i \in [n]$
  $\mathsf{hk} \leftarrow \mathsf{HGen}(1^\lambda)$

  $\mathsf{PK} := (\mathsf{pk}^0, \mathsf{pk}^1, (\mathbf{a}_i)_{i\in[n]}, \mathbf{c}, \mathsf{hk})$
  $\mathsf{SK} := (\mathsf{sk}^0, \mathsf{PK})$
  return $(\mathsf{PK}, \mathsf{SK})$
```

```
$\overline{\mathsf{Encaps}}(\mathsf{PK} = (\mathsf{pk}^0, \mathsf{pk}^1, (\mathbf{a}_i)_{i\in[n]}, \mathbf{c}, \mathsf{hk}))$:

  $\forall (i, \mathsf{b}) \in [n] \times \{0,1\}$ :
    $\mathbf{r}_i^\mathsf{b} \leftarrow \{0,1\}^\lambda$,    $(\mathsf{ct}_i^\mathsf{b}, \mathbf{k}_i^\mathsf{b}) \leftarrow \mathsf{Encaps}(\mathsf{pk}^\mathsf{b}; \mathbf{r}_i^\mathsf{b})$
  $\mathbf{s} = (s_1, \ldots, s_n) \leftarrow \mathsf{K}(1^\lambda)$,    $\boldsymbol{\kappa} \leftarrow \{0,1\}^\ell$
  $\mathsf{ct}_{\mathsf{SKE}} \leftarrow \mathsf{E}(\mathbf{s}, (\mathbf{r}_i^{s_i})_{i\in[n]} \parallel \boldsymbol{\kappa})$

  $\mathbf{h} := \mathsf{H}(\mathsf{hk}, (\mathsf{ct}_i^0, \mathsf{ct}_i^1)_{i\in[n]} \parallel \mathsf{ct}_{\mathsf{SKE}})$
  $\mathbf{t}_i := \mathbf{k}_i^{s_i} + s_i \cdot (\mathbf{a}_i + \mathbf{c} \cdot \mathbf{h})$[1]          $(\forall i \in [n])$
  $\mathsf{CT} := ((\mathsf{ct}_i^0, \mathsf{ct}_i^1, \mathbf{t}_i)_{i\in[n]}, \mathsf{ct}_{\mathsf{SKE}})$
  return $(\mathsf{CT}, \boldsymbol{\kappa})$
```

Figure 2: Algorithms $\overline{\mathsf{Gen}}$ and $\overline{\mathsf{Encaps}}$.

```
$\overline{\mathsf{Decaps}}(\mathsf{SK} = (\mathsf{sk}^0, \mathsf{PK}), \mathsf{CT} = ((\mathsf{ct}_i^0, \mathsf{ct}_i^1, \mathbf{t}_i)_{i\in[n]}, \mathsf{ct}_{\mathsf{SKE}}))$:

  $\mathbf{h} := \mathsf{H}(\mathsf{hk}, (\mathsf{ct}_i^0, \mathsf{ct}_i^1)_{i\in[n]} \parallel \mathsf{ct}_{\mathsf{SKE}})$.
  For each $i \in [n]$: if $\mathsf{Decaps}(\mathsf{sk}^0, \mathsf{ct}_i^0) = \mathbf{t}_i$, set $s_i = 0$. Otherwise set $s_i = 1$.
  $((\mathbf{r}_i^{s_i})_{i\in[n]} \parallel \boldsymbol{\kappa}) := \mathsf{D}(\mathbf{s}, \mathsf{ct}_{\mathsf{SKE}})$

  If the following holds return $\boldsymbol{\kappa}$. Otherwise return $\perp$.
    $\forall i \in [n]$ :    $\mathsf{Encaps}(\mathsf{pk}^{s_i}; \mathbf{r}_i^{s_i}) = (\mathsf{ct}_i^{s_i}, \mathbf{t}_i - s_i \cdot (\mathbf{a}_i + \mathbf{c} \cdot \mathbf{h}))$
```

Figure 3: Algorithm $\overline{\mathsf{Decaps}}$.

- Set $\mathsf{PK} = (\mathsf{pk}^0, \mathsf{pk}^1, (\mathbf{a}_i)_{i\in[n]}, \mathbf{c}, \mathsf{hk})$, $\mathsf{SK} = (\mathsf{sk}^0, \mathsf{PK})$, and then compute the ciphertext $\hat{\mathsf{CT}} = ((\hat{\mathsf{ct}}_i^0, \hat{\mathsf{ct}}_i^1, \hat{\mathbf{t}}_i)_{i\in[n]}, \hat{\mathsf{ct}}_{\mathsf{SKE}})$ as follows:

  1. Sample $(\mathsf{pk}^\mathsf{b}, \mathsf{sk}^\mathsf{b}) \leftarrow \mathsf{Gen}(1^\lambda)$ for $\mathsf{b} \in \{0,1\}$ and $\mathbf{c} \leftarrow \{0,1\}^{4\lambda}$.
  2. Sample $\hat{\mathbf{s}} \leftarrow \mathsf{K}(1^\lambda)$, $\hat{\boldsymbol{\kappa}}_1 \leftarrow \{0,1\}^\ell$, $\hat{\mathbf{r}}_i^\mathsf{b} \leftarrow \{0,1\}^{4\lambda}$ $\big(\forall (i, \mathsf{b}) \in [n] \times \{0,1\}\big)$.
  3. Compute $\hat{\mathsf{ct}}_{\mathsf{SKE}} \leftarrow \mathsf{E}(\hat{\mathbf{s}}, (\hat{\mathbf{r}}_i^{\hat{s}_i})_{i\in[n]} \parallel \hat{\boldsymbol{\kappa}}_1)$.
  4. Compute $(\hat{\mathsf{ct}}_i^\mathsf{b}, \hat{\mathbf{k}}_i^\mathsf{b}) \leftarrow \mathsf{Encaps}(\mathsf{pk}^\mathsf{b}; \hat{\mathbf{r}}_i^\mathsf{b})$ for $(i, \mathsf{b}) \in [n] \times \{0,1\}$.
  5. Compute $\mathsf{hk} \leftarrow \mathsf{HGen}(1^\lambda)$ and $\hat{\mathbf{h}} = \mathsf{H}(\mathsf{hk}, (\hat{\mathsf{ct}}_i^0, \hat{\mathsf{ct}}_i^1)_{i\in[n]} \parallel \hat{\mathsf{ct}}_{\mathsf{SKE}})$.
  6. Sample $\mathbf{a}_i \leftarrow \{0,1\}^{4\lambda}$ for $i \in [n]$.
  7. Compute $\hat{\mathbf{t}}_i = \hat{\mathbf{k}}_i^{\hat{s}_i} + \hat{s}_i \cdot (\mathbf{a}_i + \mathbf{c} \cdot \hat{\mathbf{h}})$ for $i \in [n]$.

- Sample a random key $\hat{\boldsymbol{\kappa}}_0 \leftarrow \{0,1\}^\ell$ and a bit $b \leftarrow \{0,1\}$, and run $\mathcal{A}(\mathsf{PK}, \hat{\mathsf{CT}}, \hat{\boldsymbol{\kappa}}_b)$. The adversary $\mathcal{A}$ may now start making quantum decapsulation queries.

- Decapsulation queries are answered as follows. We describe the response for any ciphertext $|\mathsf{CT}\rangle = |((\mathsf{ct}_i^0, \mathsf{ct}_i^1, \mathbf{t}_i)_{i\in[n]}, \mathsf{ct}_{\mathsf{SKE}})\rangle$ in the computational basis; the response to ciphertexts in a superposition follows in a standard way. If $\mathsf{CT} = \hat{\mathsf{CT}}$ output $\perp$. If $\mathsf{Decaps}(\mathsf{sk}^0, \mathsf{ct}_i^0) = \mathbf{t}_i$, set $s_i = 0$. Else, set $s_i = 1$ (for each $i \in [n]$). Set $\mathbf{h} = \mathsf{H}(\mathsf{hk}, (\mathsf{ct}_i^0, \mathsf{ct}_i^1)_{i\in[n]} \parallel \mathsf{ct}_{\mathsf{SKE}})$. Next, compute $((\mathbf{r}_i^{s_i})_{i\in[n]} \parallel \boldsymbol{\kappa}) := \mathsf{D}(\mathbf{s}, \mathsf{ct}_{\mathsf{SKE}})$. If for each $i \in [n]$ it holds that $\mathsf{Encaps}(\mathsf{pk}^{s_i}; \mathbf{r}_i^{s_i}) = (\mathsf{ct}_i^{s_i}, \mathbf{t}_i - s_i \cdot (\mathbf{a}_i + \mathbf{c} \cdot \mathbf{h}))$, return $\boldsymbol{\kappa}$. Otherwise return $\perp$.

- $\mathcal{A}$ finally outputs a bit $b' \in \{0,1\}$.

[1] The arithmetic is done over GF($2^{4\lambda}$) and $\mathbf{h}$ is interpreted as an element of $\{0,1\}^{4\lambda}$.

For the sake of convenience, we define the following sets:

$$S_0 := \{j \in [n] \mid \hat{s}_j = 0\}, \qquad S_1 := [n] \setminus S_0.$$

Game 2: We modify the decapsulation oracle as follows: if a ciphertext $|\mathsf{CT}\rangle = |((\mathsf{ct}_i^0, \mathsf{ct}_i^1, \mathbf{t}_i)_{i\in[n]}, \mathsf{ct}_{\mathsf{SKE}})\rangle$ satisfies $\mathbf{h} = \mathsf{H}(\mathsf{hk}, (\mathsf{ct}_i^0, \mathsf{ct}_i^1)_{i\in[n]} \parallel \mathsf{ct}_{\mathsf{SKE}}) = \hat{\mathbf{h}}$, the modified oracle returns $\perp$.

Game 3: We modify how $\mathbf{a}_i$ for the positions $i \in S_0$ are generated: for every $i \in S_0$, we generate $\mathbf{a}_i$ as $\mathbf{a}_i = \hat{\mathbf{k}}_i^0 - \hat{\mathbf{k}}_i^1 - \mathbf{c} \cdot \hat{\mathbf{h}}$.

Game 4: We modify the decapsulation oracle as follows: if a (non-challenge) ciphertext satisfies $\mathbf{h} = \mathsf{H}(\mathsf{hk}, (\mathsf{ct}_i^0, \mathsf{ct}_i^1)_{i\in[n]} \parallel \mathsf{ct}_{\mathsf{SKE}}) = \hat{\mathbf{h}}$, return $\perp$ (same as in Games 2 and 3). Otherwise, the quantum oracle uses an alternative decapsulation algorithm $\overline{\mathsf{AltDecaps}}$ and an alternative secret key SK' described below.

$\overline{\mathsf{AltDecaps}}$ takes $\mathsf{SK}' := (\mathsf{sk}^1, \mathsf{PK})$ and CT as input, and proceeds identically to $\overline{\mathsf{Decaps}}(\mathsf{SK}, \mathsf{CT})$ except when computing s_i. We instead do the following:[1]

$$\forall i \in [n]: \qquad s_i = \begin{cases} 1 & \text{if } \mathsf{Decaps}(\mathsf{sk}^1, \mathsf{ct}_i^1) = \mathbf{t}_i - \mathbf{a}_i - \mathbf{c} \cdot \mathbf{h}, \\ 0 & \text{otherwise.} \end{cases}$$

Game 5: We modify how $\mathbf{a}_i$ for the positions $i \in S_1$ are generated. For every $i \in S_1$, we generate $\mathbf{a}_i$ as $\mathbf{a}_i = \hat{\mathbf{k}}_i^0 - \hat{\mathbf{k}}_i^1 - \mathbf{c} \cdot \hat{\mathbf{h}}$. Due to this change, for every $i \in [n]$ we have $\hat{\mathbf{t}}_i = \hat{\mathbf{k}}_i^0$, irrespective of whether $\hat{s}_i$ is 0 or 1. Thus, in this game, only the value of $\hat{\mathsf{ct}}_{\mathsf{SKE}}$ is dependent on $\hat{\mathbf{s}}$.

Game 6: In this game, we generate $\hat{\mathsf{ct}}_{\mathsf{SKE}}$ in the challenge ciphertext $\hat{\mathsf{CT}}$ as $\hat{\mathsf{ct}}_{\mathsf{SKE}} \leftarrow \mathsf{E}(\hat{\mathbf{s}}, 0^{n\cdot\lambda+\ell})$. In this game, $\hat{\mathsf{ct}}_{\mathsf{SKE}}$ has no information on the bit b.

We now define $W^{(j)}$, for $j \in [6]$, to be the event when $\mathcal{A}$ succeeds in guessing the bit b (i.e., $b' = b$) in Game j. By definition, we have

$$\mathbf{Adv}^{\mathrm{qCCA}}_{\overline{\mathsf{KEM}},\mathcal{A}} = \big|\Pr[W^{(1)}] - 1/2\big|.$$

Quantum CCA security of $\overline{\mathsf{KEM}}$ follows from the following lemmas, which will be proved subsequently. □

Remark 4.2. In the following lemmas, we argue the indistinguishability of certain hybrids. We remark that the proofs of these lemmas are (almost) identical to those of [KMT19] and we briefly mention them to provide more context. The main technical part (namely Lemma 4.8) is where the proof significantly differs from its classical counterpart, which will appear subsequently.

Lemma 4.3. $|\Pr[W^{(2)}] - \Pr[W^{(3)}]| = 2 \cdot \mathbf{Adv}^{\mathrm{mCPA}}_{\mathsf{KEM},n,\mathcal{B}^3_{cpa}}$ *for some QPT adversary* $\mathcal{B}^3_{cpa}$.

Sketch. In [KMT19, Lemma 2], an (essentially) equivalent result is shown but in the context of CCA security of $\overline{\mathsf{KEM}}$. In their reduction, a PPT adversary breaks CPA security of KEM by simulating (the CCA analog of) Games 2 and 3 towards the underlying CCA adversary for $\overline{\mathsf{KEM}}$. It can be easily verified that their reduction can also be extended to the qCCA setting, because one can simulate the decapsulation oracles of Games 2 and 3 in quantum superposition given access to the secret key SK. As mentioned in Section 2, note that *any* function (in our case, $\overline{\mathsf{Decaps}}(\mathsf{SK}, \cdot)$) that has an efficient classical algorithm can also be implemented efficiently as a quantum-accessible oracle. □

Lemma 4.4. $|\Pr[W^{(3)}] - \Pr[W^{(4)}]| \le 2\varepsilon + n \cdot 2^{-\lambda+1}$.

Sketch. In [KMT19, Lemma 3], it is shown that unless the public key PK generated at the start of the CCA game for $\overline{\mathsf{KEM}}$ is "bad" in a certain sense, the *classical* decapsulation oracles in (the CCA analogs of) Games 3 and 4 (namely, $\overline{\mathsf{Decaps}}(\mathsf{SK}, \cdot)$ and $\overline{\mathsf{AltDecaps}}(\mathsf{SK}', \cdot)$) are *identical*. It was also shown in [KMT19, Lemma 3] that the probability of choosing such a "bad" PK is bounded by $2\varepsilon + n \cdot 2^{-\lambda+1}$.

The analysis in [KMT19, Lemma 3] also applies to the qCCA setting. This is because the distribution of "bad" PKs at the start of the (q)CCA game (for $\overline{\mathsf{KEM}}$) is not in any way affected by the fact that whether the adversary $\mathcal{A}$ has quantum access to the corresponding decapsulation oracle in the rest of the game. Based on [KMT19, Lemma 3], if PK is not "bad’,’ then for *any* ciphertext CT we must have $\overline{\mathsf{Decaps}}(\mathsf{SK}, \mathsf{CT}) = \overline{\mathsf{AltDecaps}}(\mathsf{SK}', \mathsf{CT})$; this means that provided PK is not "bad," the quantum decapsulation oracles in Games 3 and 4 are *identical*. □

[1] Note that we no longer require sk^0 in this modified decapsulation oracle.

Lemma 4.5. *$|\Pr[W^{(4)}] - \Pr[W^{(5)}]| = 2 \cdot \mathbf{Adv}^{\text{mCPA}}_{\mathsf{KEM},n,\mathcal{B}^4_{cpa}}$ for some QPT adversary $\mathcal{B}^4_{cpa}$.*

Sketch. A similar reasoning to that of Lemma 4.3 applies here as well. □

Lemma 4.6. *$|\Pr[W^{(5)}] - \Pr[W^{(6)}]| = 2 \cdot \mathbf{Adv}^{\text{KDM}}_{\mathsf{SKE},\mathcal{P},\mathcal{B}_{kdm}}$ for some QPT adversary $\mathcal{B}_{kdm}$ that makes a single KDM query.*

Sketch. In [KMT19, Lemma 5], an (essentially) equivalent result is shown for CCA security of $\overline{\mathsf{KEM}}$, wherein a PPT adversary breaks the one-time KDM security of SKE by simulating (the CCA analog of) Games 5 and 6 towards the underlying CCA adversary for $\overline{\mathsf{KEM}}$. By a simple extension to the qCCA setting, it is easy to see that the reduction can also simulate the decapsulation oracles of Games 5 and 6 in quantum superposition towards $\mathcal{A}$ as it generates the corresponding secret key SK' by itself. (Note that the "alternative" secret key SK' is used to compute $\overline{\mathsf{AltDecaps}}(\mathsf{SK}', \cdot)$ in Games 5 and 6.) □

Lemma 4.7. *$\Pr[W^{(6)}] = 1/2$.*

Sketch. In Game 6, the view of $\mathcal{A}$ is completely independent of the bit b (irrespective of whether it has quantum access to the decapsulation oracle). □

We now focus on the main ingredient of the proof, which is to bound the quantity $|\Pr[W^{(1)}] - \Pr[W^{(2)}]|$. As mentioned earlier, here is where the proof differs significantly from its *classical* counterpart in [KMT19, Lemma 1]. Namely, in the qCCA setting, we argue about the *quantum* indistinguishability of the decapsulation oracles in Games 1 and 2 using Lemma 2.1, while following a similar "deferred analysis" approach as in [KMT19].

Lemma 4.8. *There exist QPT adversaries $\mathcal{B}_{tcr}$, $\mathcal{B}^1_{cpa}$, $\mathcal{B}^2_{cpa}$, and $\mathcal{B}'_{cpa}$ satisfying*

$$|\Pr[W^{(1)}] - \Pr[W^{(2)}]| \leq 2q\sqrt{\mathsf{Adv} + 10\varepsilon + n \cdot 2^{-4\lambda+1} + n \cdot 2^{-\lambda+1}}$$

where $\mathsf{Adv} = \mathbf{Adv}^{\text{TCR}}_{\mathsf{Hash},\mathcal{B}_{tcr}} + 2 \cdot (\mathbf{Adv}^{\text{mCPA}}_{\mathsf{KEM},n,\mathcal{B}^1_{cpa}} + \mathbf{Adv}^{\text{mCPA}}_{\mathsf{KEM},n,\mathcal{B}^2_{cpa}} + \mathbf{Adv}^{\text{mCPA}}_{\mathsf{KEM},n,\mathcal{B}'_{cpa}})$.

Proof. Following the terminology of [KMT19], we call a decapsulation query $\mathsf{CT} = ((\mathsf{ct}^0_i, \mathsf{ct}^1_i, \mathbf{t}_i)_{i\in[n]}, \mathsf{ct}_{\mathsf{SKE}})$ of $\mathcal{A}$ in Game j (for some $j \in [4]$) *hash-bad* if

$$\mathbf{h} = \mathsf{H}(\mathsf{hk}, (\mathsf{ct}^0_i, \mathsf{ct}^1_i)_{i\in[n]} \parallel \mathsf{ct}_{\mathsf{SKE}}) = \hat{\mathbf{h}} \quad \text{and} \quad \overline{\mathsf{Decaps}}(\mathsf{SK}, \mathsf{CT}) \neq \perp$$

such that $\mathsf{CT} \neq \hat{\mathsf{CT}}$. Observe that the outputs of decapsulation oracles of Games 1 and 2 differ *exactly* in these hash-bad queries. We also categorize a hash-bad decapsulation query $\mathsf{CT} = ((\mathsf{ct}^0_i, \mathsf{ct}^1_i, \mathbf{t}_i)_{i\in[n]}, \mathsf{ct}_{\mathsf{SKE}})$ as follows:

- Type 1: $(\mathsf{ct}^0_i, \mathsf{ct}^1_i)_{i\in[n]} \parallel \mathsf{ct}_{\mathsf{SKE}} \neq (\hat{\mathsf{ct}}^0_i, \hat{\mathsf{ct}}^1_i)_{i\in[n]} \parallel \hat{\mathsf{ct}}_{\mathsf{SKE}}$
- Type 2: $(\mathsf{ct}^0_i, \mathsf{ct}^1_i)_{i\in[n]} \parallel \mathsf{ct}_{\mathsf{SKE}} = (\hat{\mathsf{ct}}^0_i, \hat{\mathsf{ct}}^1_i)_{i\in[n]} \parallel \hat{\mathsf{ct}}_{\mathsf{SKE}}$

We rely on Lemma 2.1 to bound the term $|\Pr[W^{(1)}] - \Pr[W^{(2)}]|$ as follows. First, let A be a quantum oracle algorithm which receives as input a public key $\mathsf{PK} = (\mathsf{pk}^0, \mathsf{pk}^1, (\mathbf{a}_i)_{i\in[n]}, \mathbf{c}, \mathsf{hk})$, the "real" encapsulated key $\hat{\boldsymbol{\kappa}}_1 \in \{0,1\}^\ell$ and a challenge ciphertext $\hat{\mathsf{CT}} = ((\hat{\mathsf{ct}}^0_i, \hat{\mathsf{ct}}^1_i, \hat{\mathbf{t}}_i)_{i\in[n]}, \hat{\mathsf{ct}}_{\mathsf{SKE}})$ as generated in Game 1 above. A has quantum access either to the decapsulation oracle $G := \overline{\mathsf{Decaps}}(\mathsf{SK}, \cdot)$ used in Game 1 (which also rejects ciphertexts equal to $\hat{\mathsf{CT}}$) or to the modified decapsulation oracle H used in Game 2. A proceeds to simulate either Game 1 or 2 towards $\mathcal{A}$ as follows: A samples a random key $\hat{\boldsymbol{\kappa}}_0 \leftarrow \{0,1\}^\ell$ and a bit $b \leftarrow \{0,1\}$, and forwards $(\mathsf{PK}, \hat{\mathsf{CT}}, \hat{\boldsymbol{\kappa}}_b)$ to $\mathcal{A}$. Then A responds to $\mathcal{A}$'s quantum decapsulation queries using its oracle (i.e., G or H). A outputs 1 iff $b' = b$.

By applying Lemma 2.1, it follows that $\Pr[W^{(1)}] = \Pr[1 \leftarrow A^G(z)]$ and $\Pr[W^{(2)}] = \Pr[1 \leftarrow A^H(z)]$. Thus we have $|\Pr[W^{(1)}] - \Pr[W^{(2)}]| \leq 2q\sqrt{P_{\text{guess}}}$ where P_{guess} is essentially the probability of the event when measurement of a random quantum decapsulation query made by $\mathcal{A}$ in Game 2 would result in a *hash-bad* ciphertext CT ($\neq \hat{\mathsf{CT}}$) defined above, i.e., $G(\mathsf{CT}) \neq H(\mathsf{CT})$.

For $(j, \mathsf{b}) \in [4] \times [2]$, let $M_j^{(\mathsf{b})}$ be the event that the measurement of a random i-th quantum decapsulation query made by $\mathcal{A}$ in Game j (where $i \leftarrow [q]$) results in a type b hash-bad ciphertext. (Note that $\Pr[M_2^{(1)}] + \Pr[M_2^{(2)}] = P_{\text{guess}}$.)

Based on Lemma 2.1, we have

$$|\Pr[W^{(1)}] - \Pr[W^{(2)}]| \leq 2q\sqrt{\Pr[M_2^{(1)}] + \Pr[M_2^{(2)}]}.$$

Observe that a type 1 hash-bad query can be used to break the target collision resistance of Hash. To see this, we construct a QPT adversary $\mathcal{B}_{tcr}$ such that $\Pr[M_2^{(1)}] = \mathbf{Adv}^{\mathsf{TCR}}_{\mathsf{Hash}, \mathcal{B}_{tcr}}$. First $\mathcal{B}_{tcr}$ generates the values $(\hat{\mathsf{ct}}_i^0, \hat{\mathsf{ct}}_i^1)_{i \in [n]}$ and $\hat{\mathsf{ct}}_{\mathsf{SKE}}$ by itself as in Game 2 and forwards $(\hat{\mathsf{ct}}_i^0, \hat{\mathsf{ct}}_i^1)_{i \in [n]} \parallel \hat{\mathsf{ct}}_{\mathsf{SKE}}$ to its TCR challenger (with respect to Hash). After obtaining a key hk from its challenger, $\mathcal{B}_{tcr}$ forwards the public key PK (which includes hk) along with the values $\hat{\mathsf{CT}}$, $\hat{\boldsymbol{\kappa}}_b$ to $\mathcal{A}$. Next, $\mathcal{B}_{tcr}$ samples $i \leftarrow [q]$ and proceeds to simulate the quantum decapsulation oracle of Game 2 towards $\mathcal{A}$. Observe that this is possible because $\mathcal{B}_{tcr}$ generates the secret key SK by itself. $\mathcal{B}_{tcr}$ then measures the i-th decapsulation query of $\mathcal{A}$ and forwards the measurement to its TCR challenger.

It remains to bound $\Pr[M_2^{(2)}]$. For type 2 hash-bad queries we have $\mathsf{CT} \neq \hat{\mathsf{CT}}$, and hence there exists a position $j \in [n]$ such that $\mathbf{t}_j \neq \hat{\mathbf{t}}_j$. For a type 2 hash-bad ciphertext $\mathsf{CT} = ((\hat{\mathsf{ct}}_i^0, \hat{\mathsf{ct}}_i^1, \mathbf{t}_i)_{i \in [n]}, \hat{\mathsf{ct}}_{\mathsf{SKE}})$, we define $S_{\mathsf{CT}} = \{j \in [n] \mid \mathbf{t}_j \neq \hat{\mathbf{t}}_j\}$. As in the proof of [KMT19, Lemma 1], we have the following for positions $i \in S_{\mathsf{CT}}$ *conditioned* on pk^0 and pk^1 not resulting in decapsulation errors.

- If $\hat{s}_i = 0$ (i.e., $i \in S_{\mathsf{CT}} \cap S_0$), then $\mathbf{t}_i - \mathbf{a}_i - \mathbf{c} \cdot \hat{\mathbf{h}} = \hat{\mathbf{k}}_i^1$ where $\hat{\mathbf{k}}_i^1$ is the encapsulated key corresponding to $\hat{\mathsf{ct}}_i^1$.
- If $\hat{s}_i = 1$ (i.e., $i \in S_{\mathsf{CT}} \cap S_1$), then $\mathbf{t}_i = \hat{\mathbf{k}}_i^0$ where $\hat{\mathbf{k}}_i^0$ is the encapsulated key corresponding to $\hat{\mathsf{ct}}_i^0$.

Consider the following categorization of type 2 queries into two sub-types:

- Type 2a: There exists a position $i \in S_{\mathsf{CT}} \cap S_0$.
- Type 2b: There exists a position $i \in S_{\mathsf{CT}} \cap S_1$.

For $j \in \{2, 3, 4\}$ and $\mathsf{b} \in \{2a, 2b\}$, let $M_j^{(\mathsf{b})}$ be the event that the measurement of a random i-th quantum query of $\mathcal{A}$ in Game j (where $i \leftarrow [q]$) results in a type b hash-bad decapsulation query. First, we have $\Pr[M_2^{(2)}] \leq \Pr[M_2^{(2a)}] + \Pr[M_2^{(2b)}]$. Towards bounding $\Pr[M_2^{(2)}]$, we first show that there exists a QPT adversary $\mathcal{B}^1_{cpa}$ that breaks the CPA security of KEM and satisfies

$$\Pr[M_2^{(2a)}] \leq 2 \cdot \mathbf{Adv}^{\mathsf{mCPA}}_{\mathsf{KEM}, n, \mathcal{B}^1_{cpa}} + 4\varepsilon + n \cdot 2^{-4\lambda}.$$

The description of $\mathcal{B}^1_{cpa}$ is as follows: on input $(\mathsf{pk}', (\hat{\mathsf{ct}}_i', \hat{\mathbf{k}}_{i,\beta}'))$ where the bit β is $\mathcal{B}^1_{cpa}$'s challenge bit, $\mathcal{B}^1_{cpa}$ first runs $\hat{\mathsf{s}} \leftarrow \mathsf{K}(1^\lambda)$, and sets $\mathsf{pk}^1 := \mathsf{pk}'$ and $\hat{\mathsf{ct}}_i^1 := \hat{\mathsf{ct}}_i'$ for the positions $i \in S_0$. Next, $\mathcal{B}^1_{cpa}$ generates the remaining values of PK, SK, $\hat{\mathsf{CT}}$, and $\hat{\boldsymbol{\kappa}}_b$ by itself as in Game 2 and forwards $(\mathsf{PK}, \hat{\mathsf{CT}}, \hat{\boldsymbol{\kappa}}_b)$ to $\mathcal{A}$. Next, $\mathcal{B}^1_{cpa}$ samples $i \leftarrow [q]$ and proceeds to simulate the quantum decapsulation oracle of Game 2 towards $\mathcal{A}$ until the i-th decapsulation query; $\mathcal{B}^1_{cpa}$ measures the i-th query and checks if it is a type 2a hash-bad query CT.

- If the measurement results in a type 2a query $\mathsf{CT} = ((\hat{\mathsf{ct}}_i^0, \hat{\mathsf{ct}}_i^1, \mathbf{t}_i)_{i \in [n]}, \hat{\mathsf{ct}}_{\mathsf{SKE}})$, $\mathcal{B}^1_{cpa}$ checks if there is a position $i \in S_{\mathsf{CT}} \cap S_0$ such that $\mathbf{t}_i - \mathbf{a}_i - \mathbf{c} \cdot \hat{\mathbf{h}} = \hat{\mathbf{k}}_{i,\beta}'$. If so, $\mathcal{B}^1_{cpa}$ sets $\beta' = 1$. Otherwise, it sets $\beta' = 0$.
- If the measurement does not result in a type 2a query, then $\mathcal{B}^1_{cpa}$ sets $\beta' = 0$.

Finally, $\mathcal{B}^1_{cpa}$ terminates with the output β'.

Observe that $\mathcal{B}^1_{cpa}$ properly simulates Game 2 towards $\mathcal{A}$ (regardless of the challenge bit β) until the i-th quantum decapsulation query. Hence, the probability that the measurement of the i-th decapsulation query made by $\mathcal{A}$ results in a type 2a query is $\Pr[M_2^{(2a)}]$. Now recall from the above observation that conditioned on pk^0 and pk^1 not resulting in decapsulation errors, if the measurement results in a type 2a query $\mathsf{CT} = ((\hat{\mathsf{ct}}_i^0, \hat{\mathsf{ct}}_i^1, \mathbf{t}_i)_{i \in [n]}, \hat{\mathsf{ct}}_{\mathsf{SKE}})$, then

$\mathbf{t}_i - \mathbf{a}_i - \mathbf{c} \cdot \hat{\mathbf{h}} = \hat{\mathbf{k}}_i^1$ holds for positions $i \in S_{\mathsf{CT}} \cap S_0$. Since $\mathcal{B}_{cpa}^1$ embeds each of its given challenge ciphertexts $\hat{\mathsf{ct}}_i'$ as $\hat{\mathsf{ct}}_i^1$ for the positions $i \in S_0$, if $\beta = 1$ (i.e., the keys $\hat{\mathbf{k}}_{i,\beta}'$ given to $\mathcal{B}_{cpa}^1$ are "real" encapsulated keys with respect to $\hat{\mathsf{ct}}_i'$), then we have $\hat{\mathbf{k}}_i^1 = \hat{\mathbf{k}}_{i,1}'$ for the positions $i \in S_0$. Thus, if $\beta = 1$, then there is at least one position $i \in S_{\mathsf{CT}} \cap S_0$ for which $\mathbf{t}_i - \mathbf{a}_i - \mathbf{c} \cdot \hat{\mathbf{h}} = \hat{\mathbf{k}}_i^1 = \hat{\mathbf{k}}_{i,1}'$ holds, and $\mathcal{B}_{cpa}^1$ outputs $\beta' = 1$ (conditioned on pk^0 and pk^1 not resulting in decapsulation errors). If we denote E to be the event that the sampled public keys pk^0 and pk^1 do result in decapsulation errors, then we have $\Pr[\beta' = 1 | \beta = 1 \wedge \neg E] = \Pr[M_2^{(2a)} | \neg E]$.

Similarly, if $\beta = 0$, then the keys $\hat{\mathbf{k}}_{i,0}'$ for every $i \in [n]$ are chosen uniformly from $\{0,1\}^{4\lambda}$ and are completely independent of $\mathcal{A}$'s view. Thus, the probability that there exists $i \in S_{CT} \cap S_0$ for which $\mathbf{t}_i - \mathbf{a}_i - \mathbf{c} \cdot \hat{\mathbf{h}} = \hat{\mathbf{k}}_{i,0}'$ holds (and $\mathcal{B}_{cpa}^1$ outputs $\beta' = 1$ when $\beta = 0$) is at most $n \cdot 2^{-4\lambda}$ by a union bound. Therefore, we have $\Pr[\beta' = 1 | \beta = 0] \leq n \cdot 2^{-4\lambda}$. It follows that

$$2 \cdot \mathbf{Adv}_{\mathsf{KEM},n,\mathcal{B}_{cpa}^1}^{\mathsf{mCPA}} = |\Pr[\beta' = 1 | \beta = 1] - \Pr[\beta' = 1 | \beta = 0]|.$$

Since KEM is ε-all-keys correct, we have $\Pr[E] \leq 2\varepsilon$. By a routine calculation we have

$$\begin{aligned} \Pr[\beta' = 1 | \beta = 1] &\geq \Pr[\beta' = 1 | \beta = 1 \wedge \neg E] - 2\varepsilon = \Pr[M_2^{(2a)} | \neg E] - 2\varepsilon \\ &\geq (\Pr[M_2^{(2a)}] - 2\varepsilon) - 2\varepsilon = \Pr[M_2^{(2a)}] - 4\varepsilon, \end{aligned}$$

and hence it follows that $2 \cdot \mathbf{Adv}_{\mathsf{KEM},n,\mathcal{B}_{cpa}^1}^{\mathsf{mCPA}} \geq \Pr[M_2^{(2a)}] - 4\varepsilon - n \cdot 2^{-4\lambda}$.

Next, we show how to bound $\Pr[M_2^{(2b)}]$. Here we use a "deferred analysis" approach, as in the proof of [KMT19, Lemma 1]. By triangle inequality we have

$$\Pr[M_2^{(2b)}] \leq \sum_{j \in \{2,3\}} \left| \Pr[M_j^{(2b)}] - \Pr[M_{j+1}^{(2b)}] \right| + \Pr[M_4^{(2b)}].$$

Here, with essentially the same argument as in the proof of Lemma 4.3, we have

$$|\Pr[M_2^{(2b)}] - \Pr[M_3^{(2b)}]| = 2 \cdot \mathbf{Adv}_{\mathsf{KEM},n,\mathcal{B}_{cpa}^2}^{\mathsf{mCPA}}$$

for a QPT adversary $\mathcal{B}_{cpa}^2$. To be more specific, in the reduction, $\mathcal{B}_{cpa}^2$ samples a query index $i \leftarrow [q]$ and runs in exactly the same way as $\mathcal{B}_{cpa}^3$ in the proof of Lemma 4.3, until the i-th query made by $\mathcal{A}$. $\mathcal{B}_{cpa}^2$ instead measures the i-th query and returns 1 if and only if the measurement results in a type 2b hash-bad query CT, which can be checked since $\mathcal{B}_{cpa}^2$ has access to sk^0 (to check if $\overline{\mathsf{Decaps}}(\mathsf{SK}, \mathsf{CT}) \neq \bot$).

Similarly, with the same argument as in the proof of Lemma 4.4, we have $|\Pr[M_3^{(2b)}] - \Pr[M_4^{(2b)}]| \leq 2\varepsilon + n \cdot 2^{-\lambda+1}$. Finally, we show that there exists a QPT adversary $\mathcal{B}_{cpa}'$ that breaks the CPA security of KEM and satisfies

$$\Pr[M_4^{(2a)}] \leq 2 \cdot \mathbf{Adv}_{\mathsf{KEM},n,\mathcal{B}_{cpa}'}^{\mathsf{mCPA}} + 4\varepsilon + n \cdot 2^{-4\lambda}.$$

The description of $\mathcal{B}_{cpa}'$ is quite similar to that of $\mathcal{B}_{cpa}^1$ above: on input $(\mathsf{pk}', (\hat{\mathsf{ct}}_i', \hat{\mathbf{k}}_{i,\beta}'))$, $\mathcal{B}_{cpa}'$ first runs $\hat{\mathsf{s}} \leftarrow \mathsf{K}(1^\lambda)$, and sets $\mathsf{pk}^0 = \mathsf{pk}'$ and $\hat{\mathsf{ct}}_i^0 = \hat{\mathsf{ct}}_i'$ for positions $i \in S_1$. Next, $\mathcal{B}_{cpa}'$ generates the remaining values of PK, SK, $\hat{\mathsf{CT}}$, and $\hat{\boldsymbol{\kappa}}_b$ by itself as in Game 4, and forwards $(\mathsf{PK}, \hat{\mathsf{CT}}, \hat{\boldsymbol{\kappa}}_b)$ to $\mathcal{A}$. Next, $\mathcal{B}_{cpa}'$ samples $i \leftarrow [q]$ and simulates the quantum decapsulation oracle with respect to $\overline{\mathsf{AltDecaps}}(\mathsf{SK}', \cdot)$ in Game 4 towards $\mathcal{A}$ (note that $\mathcal{B}_{cpa}'$ has access to sk^1 which is sufficient to compute $\overline{\mathsf{AltDecaps}}(\mathsf{SK}', \cdot)$) until the i-th decapsulation query. $\mathcal{B}_{cpa}'$ measures the i-th query (and does *not* check if it is a type 2b query since $\mathcal{B}_{cpa}'$ does not have sk^0). Let the measured query be $\mathsf{CT} = ((\mathsf{ct}_i^0, \mathsf{ct}_i^1, \mathbf{t}_i)_{i \in [n]}, \mathsf{ct}_{\mathsf{SKE}})$. Finally, $\mathcal{B}_{cpa}'$ checks if there exists a position $i \in S_{\mathsf{CT}} \cap S_1$ such that $\mathbf{t}_i = \hat{\mathbf{k}}_{i,\beta}'$. If so, $\mathcal{B}_{cpa}'$ outputs $\beta' = 1$. Otherwise it outputs $\beta' = 0$.

As in the analysis of $\mathcal{B}_{cpa}^1$, observe that $\mathcal{B}_{cpa}'$ simulates Game 4 towards $\mathcal{A}$ (regardless of the challenge bit β) until the i-th quantum decapsulation query. Hence, the probability that the measurement of the i-th decapsulation query of $\mathcal{A}$ results in a type 2b query is $\Pr[M_2^{(2b)}]$. Recall from the above observation that conditioned on pk^0 and pk^1 not resulting in decapsulation errors, if the measurement results in a type 2b query $\mathsf{CT} = ((\hat{\mathsf{ct}}_i^0, \hat{\mathsf{ct}}_i^1, \mathbf{t}_i)_{i\in[n]}, \hat{\mathsf{ct}}_{\mathsf{SKE}})$, then $\mathbf{t}_i = \hat{\mathbf{k}}_i^0$ holds for positions $i \in S_{\mathsf{CT}} \cap S_1$. Since $\mathcal{B}'_{cpa}$ embeds each of its given challenge ciphertexts $\hat{\mathsf{ct}}'_i$ as $\hat{\mathsf{ct}}_i^0$ for the positions $i \in S_1$, if $\beta = 1$ then we have $\hat{\mathbf{k}}_i^0 = \hat{\mathbf{k}}'_{i,1}$ for the positions $i \in S_1$. Thus, if $\beta = 1$ then there is at least one position $i \in S_{\mathsf{CT}} \cap S_1$ for which $\mathbf{t}_i = \hat{\mathbf{k}}_i^0 = \hat{\mathbf{k}}'_{i,1}$ holds, and $\mathcal{B}^1_{cpa}$ outputs $\beta' = 1$ (conditioned on pk^0 and pk^1 not resulting in decapsulation errors). Let E denote the event that the sampled KEM public keys pk^0 and pk^1 do result in decapsulation errors, then we have $\Pr[\beta' = 1 | \beta = 1 \wedge \neg E] = \Pr[M_4^{(2b)} | \neg E]$.

Similarly, if $\beta = 0$, then the keys $\hat{\mathbf{k}}'_{i,0}$ for every $i \in [n]$ are chosen uniformly from $\{0,1\}^{4\lambda}$ and are completely independent of $\mathcal{A}$'s view. Thus, the probability that there exists $i \in S_{\mathsf{CT}} \cap S_1$ for which $\mathbf{t}_i = \hat{\mathbf{k}}'_{i,0}$ holds (and $\mathcal{B}'_{cpa}$ outputs $\beta' = 1$ when $\beta = 0$) is at most $n \cdot 2^{-4\lambda}$ by a union bound. Therefore, we have $\Pr[\beta' = 1 | \beta = 0] \leq n \cdot 2^{-4\lambda}$.

By a routine calculation as in the one for $\mathcal{B}^1_{cpa}$'s advantage above, we have[1]

$$2 \cdot \mathbf{Adv}^{\text{mCPA}}_{\mathsf{KEM},n,\mathcal{B}'_{cpa}} \geq \Pr[M_4^{(2b)}] - 4\varepsilon - n \cdot 2^{-4\lambda}. \qquad \square$$

Next, we prove that qCCA-secure KEM implies qCCA-secure PKE by showing that the hybrid encryption (KEM-DEM) framework of Cramer and Shoup [CS03] results in a qCCA-secure PKE if a qCCA-secure KEM is composed with a *classical* (post-quantum) one-time authenticated encryption scheme.[2]

5 KEM-DEM Composition and Quantum CCA-Secure PKE

In the classical setting, it has long been known that a CCA-secure PKE can be constructed from a CCA-secure KEM along with a one-time authenticated encryption scheme (which can be constructed from any one-way function) via the hybrid encryption approach of [CS03]. However, in the quantum setting, it is not known such an approach would result in a qCCA-secure PKE. A few recent works (e.g., [XY19, LW21]) showed realizations of qCCA-secure KEM in the QROM, but they did not discuss qCCA-secure KEM (and its implications) in the *standard* model. In this section, we prove that the KEM-DEM hybrid encryption of [CS03] can also be used to construct qCCA-secure PKE provided that the underlying KEM is qCCA-secure and, perhaps somewhat surprisingly, the underlying DEM offers *classical* one-time authenticated encryption security (with respect to QPT adversaries).

We recall the PKE construction of [CS03] via the KEM-DEM composition. Let $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \mathsf{Encaps}, \mathsf{Decaps})$ be a qCCA-secure KEM and $\mathsf{DEM} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$ be a (classical) one-time authenticated encryption scheme. Consider the following PKE scheme $\mathsf{PKE}^{hyb} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$:

$\mathsf{Gen}(1^\lambda)$: Sample a pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \overline{\mathsf{Gen}}(1^\lambda)$ and output$(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$: Compute $(\mathsf{ct}, \mathsf{k}) \leftarrow \mathsf{Encaps}(\mathsf{pk})$ and $c \leftarrow \mathsf{E}(\mathsf{k}, \mathsf{m})$. Output (ct, c).

$\mathsf{Dec}(\mathsf{sk}, (\mathsf{ct}, c))$: Compute $\mathsf{k} \leftarrow \mathsf{Decaps}(\mathsf{sk}, \mathsf{ct})$. If $\mathsf{k} \neq \bot$, return $\mathsf{m} = \mathsf{D}(\mathsf{k}, c)$; otherwise, return $\bot$.

Theorem 5.1. *If* KEM *is a qCCA-secure scheme with almost-all-keys correctness and* DEM *is a (classical) one-time authenticated encryption scheme (with respect to QPT adversaries), then* PKE^{hyb} *is a qCCA-secure PKE.*

Before diving into the proof, note that in the classical setting, one can show the CCA security of PKE^{hyb} either by relying on the CCA security of DEM or the authenticated encryption security (i.e., CPA + INT-CTXT security) of DEM. In the former case, the reduction crucially relies on (classical) access to DEM's decryption oracle $\mathsf{D}(\mathsf{k}^*, \cdot)$ to answer decryption queries. So if we extend this reduction strategy to show qCCA security of PKE^{hyb}, we need to rely on the

[1] It is worth pointing out that in the bounds obtained on the classical CCA analog of $\Pr[M_4^{(2b)}]$ in [KMT19, Lemma 1], there is a $(1/q)$ multiplicative factor, since in their reduction, the CPA adversary (with respect to KEM) chooses one of $\mathcal{A}$'s decapsulation queries uniformly at random. However, we do not have such a factor in our bounds since by applying Lemma 2.1, we are already measuring one of $\mathcal{A}$'s decapsulation queries uniformly at random; i.e., this "random guessing" is accounted for in the definition of $P_{\text{guess}} = \Pr[M_4^{(2b)}]$.

[2] Such a scheme is implied by post-quantum one-way functions.

qCCA security of DEM, i.e., the reduction needs *quantum* access to $|\mathsf{D}(\mathsf{k}^*,\cdot)\rangle$ to answer quantum decryption queries. However, if we instead extend the latter reduction strategy (based on authenticated encryption security of DEM) to prove the qCCA security of PKE^{hyb}, then we can use the OW2H lemma (Lemma 2.1) to show that the *classical* security of DEM suffices.

Proof. Let $\mathcal{A}$ be any QPT adversary that breaks the quantum CCA security of PKE^{hyb} while making q quantum decryption queries with q_{pre}/q_{post} queries in the pre/post-challenge phase. Also, let KEM be ε-almost-all-keys correct for some negligible ε. Now consider the following sequence of games.

Game 1: This is essentially the same as the qCCA game except for some minor changes.[1]

- Sample $(\mathsf{pk},\mathsf{sk}) \leftarrow \overline{\mathsf{Gen}}(1^\lambda)$ and $b \leftarrow \{0,1\}$. Compute $(\mathsf{ct}^*,\mathsf{k}^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$.
- Forward pk to $\mathcal{A}$ and respond to $\mathcal{A}$'s quantum decryption queries in the normal way using the description of $\mathsf{Dec}(\mathsf{sk},\cdot)$ above.
- After receiving $(\mathsf{m}_0,\mathsf{m}_1)$ from $\mathcal{A}$, compute $c^* \leftarrow \mathsf{E}(\mathsf{k}^*,\mathsf{m}_b)$ and forward the challenge ciphertext (ct^*,c^*) to $\mathcal{A}$.
- Respond to $\mathcal{A}$'s quantum decryption queries in the normal way, but this time making sure to reject ciphertexts that are equal to (ct^*,c^*).
- $\mathcal{A}$ then terminates with an output $b' \in \{0,1\}$.

Game 2: We modify the decryption oracle as follows: for any ciphertext $|(\mathsf{ct},c)\rangle$ such that $\mathsf{ct}=\mathsf{ct}^*$, the oracle uses the key k^* *directly* to decrypt c, instead of first decapsulating ct^* to recover a session key k.

Game 3: We compute c^* in the setup as $c^* \leftarrow \mathsf{E}(\hat{\mathsf{k}},\mathsf{m}_b)$, instead of $c^* \leftarrow \mathsf{E}(\mathsf{k}^*,\mathsf{m}_b)$, for a random key $\hat{\mathsf{k}}$ independent of k^*. We also make an appropriate modification to the decryption oracle as well: given a ciphertext $|(\mathsf{ct},c)\rangle$ such that $\mathsf{ct}=\mathsf{ct}^*$, the oracle uses the key $\hat{\mathsf{k}}$ (instead of k^*) to decrypt c.

Game 4a: We modify the decryption oracle in the *pre-challenge phase* as follows: for any $|(\mathsf{ct},c)\rangle$ such that $\mathsf{ct}=\mathsf{ct}^*$, the modified oracle returns $\perp$.

Game 4b: We modify the decryption oracle in the *post-challenge phase* as follows: for any $|(\mathsf{ct},c)\rangle$ such that $\mathsf{ct}=\mathsf{ct}^*$, the modified oracle returns $\perp$.

Let $W^{(j)}$ (for $j \in [3] \cup \{4a,4b\}$) be the event that $\mathcal{A}$ succeeds in guessing the bit b (i.e., $b'=b$) in Game j. By definition, we have

$$\mathbf{Adv}^{\mathrm{qCCA}}_{\mathsf{PKE}^{hyb},\mathcal{A}} = \left|\Pr[W^{(1)}] - \frac{1}{2}\right|.$$

We now have the following in the subsequent games.

Lemma 5.2. $|\Pr[W^{(1)}] - \Pr[W^{(2)}]| \le \varepsilon$.

Proof. It's easy to see that Games 1 and 2 proceed identically unless there is a decapsulation error, i.e., $(\mathsf{ct}^*,\mathsf{k}^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$ but $\mathsf{Decaps}(\mathsf{sk},\mathsf{ct}^*) = \mathsf{k} \neq \mathsf{k}^*$. The probability of generating such an "erroneous" key pair $(\mathsf{pk},\mathsf{sk})$ is bounded by ε, because of KEM's ε-almost-all-keys correctness. □

Lemma 5.3. *There exists a QPT adversary $\mathcal{B}_{qcca}$ such that*

$$|\Pr[W^{(2)}] - \Pr[W^{(3)}]| = \mathbf{Adv}^{\mathrm{qCCA}}_{\mathsf{KEM},\mathcal{B}_{qcca}}.$$

Proof. On input $(\mathsf{pk},(\mathsf{ct}^*,\mathsf{k}^*_\beta))$ where $\beta \in \{0,1\}$ is $\mathcal{B}_{qcca}$'s challenge bit, $\mathcal{B}_{qcca}$ samples $b \leftarrow \{0,1\}$. It forwards pk to $\mathcal{A}$ and responds to $\mathcal{A}$'s quantum decryption queries in the pre-challenge phase using its *quantum* access to the decapsulation oracle $\mathsf{Decaps}(\mathsf{sk},\cdot)$ as follows: given any query $|(\mathsf{ct},c)\rangle$ in the computational basis, if $\mathsf{ct}=\mathsf{ct}^*$, then return $\mathsf{D}(\mathsf{k}^*_\beta,c)$; otherwise, compute $\mathsf{k}=\mathsf{Decaps}(\mathsf{sk},\mathsf{ct})$ using access to the corresponding decapsulation oracle and

[1] Specifically, the pair $(\mathsf{ct}^*,\mathsf{k}^*)$ is generated by running $\mathsf{Encaps}(\mathsf{pk})$ *before* $\mathcal{A}$ gets to choose a pair of messages $(\mathsf{m}_0,\mathsf{m}_1)$. However, this change does not affect $\mathcal{A}$'s view compared to the original qCCA game.

return $\mathsf{D}(\mathsf{k}, c)$. After receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, the adversary $\mathcal{B}_{qcca}$ computes $c^* \leftarrow \mathsf{E}(\mathsf{k}^*_\beta, \mathsf{m}_b)$ and forwards (ct^*, c^*) to $\mathcal{A}$. Then $\mathcal{B}_{qcca}$ proceeds to answer the rest of $\mathcal{A}$'s quantum decryption queries in the post-challenge phase as above while making sure to reject ciphertexts that are equal to (ct^*, c^*). Finally, when $\mathcal{A}$ terminates with an output $b' \in \{0, 1\}$, $\mathcal{B}_{qcca}$ outputs 1 if $b' = b$. Otherwise, it outputs 0.

Note that if $\beta = 1$, i.e., k^*_β is a "real" encapsulated key, then $\mathcal{B}_{qcca}$ perfectly simulates Game 2 towards $\mathcal{A}$. Similarly, if $\beta = 0$, i.e., k^*_β is a uniform encapsulated key independent of ct^*, then $\mathcal{B}_{qcca}$ perfectly simulates Game 3. Hence, we have $\Pr[\beta' = 1|\beta = 1] = \Pr[W^{(2)}]$ and $\Pr[\beta' = 1|\beta = 0] = \Pr[W^{(3)}]$. It follows that

$$\mathbf{Adv}^{\text{qCCA}}_{\mathsf{KEM},\mathcal{B}_{qcca}} = |\Pr[\beta' = 1|\beta = 1] - \Pr[\beta' = 1|\beta = 0]| = |\Pr[W^{(2)}] - \Pr[W^{(3)}]|. \qquad \square$$

Lemma 5.4. *There exists a QPT adversary $\mathcal{B}^1_{ctxt}$ such that*

$$|\Pr[W^{(3)}] - \Pr[W^{(4a)}]| \leq 2q_{pre}\sqrt{\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM},\mathcal{B}^1_{ctxt}}}.$$

Proof. We use Lemma 2.1 to bound $|\Pr[W^{(3)}] - \Pr[W^{(4a)}]|$. Let A be a quantum oracle algorithm which receives as input a pair $(\mathsf{pk}, \mathsf{sk})$, a KEM ciphertext ct^* where $(\mathsf{ct}^*, \mathsf{k}^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$ and a uniform DEM key $\hat{\mathsf{k}}$ as generated in Game 3 above. A also has quantum access to either the corresponding pre-challenge decryption oracle G in Game 3 (which uses $\hat{\mathsf{k}}$ to respond to ciphertexts of the form $|(\mathsf{ct}^*, c)\rangle$) or to the pre-challenge decryption oracle H in Game 4a (which rejects ciphertexts $|(\mathsf{ct}^*, c)\rangle$). A then proceeds to simulate either Game 3 or 4a towards $\mathcal{A}$ depending on whether it has access to G or H respectively as follows: A samples $b \leftarrow \{0, 1\}$. It then forwards pk to $\mathcal{A}$ and proceeds to respond to $\mathcal{A}$'s quantum decryption queries in the pre-challenge phase using its own oracle (i.e., G or H). Then after receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, the algorithm A computes $c^* \leftarrow \mathsf{E}(\hat{\mathsf{k}}, \mathsf{m}_b)$ and forwards (ct^*, c^*) to $\mathcal{A}$. Next, A proceeds to respond to $\mathcal{A}$'s quantum decryption queries in the post-challenge phase as in Game 3, this time using the keys sk and $\hat{\mathsf{k}}$ (and *not* the oracles G or H), while at the same time rejecting ciphertexts that are equal to (ct^*, c^*). Finally A outputs 1 if the output of $\mathcal{A}$ is equal to b.

Observe that $\Pr[W^{(3)}] = \Pr[1 \leftarrow A^G(z)]$ and $\Pr[W^{(4a)}] = \Pr[1 \leftarrow A^H(z)]$. By Lemma 2.1, it follows that $|\Pr[W^{(3)}] - \Pr[W^{(4a)}]| \leq 2q_{pre}\sqrt{P_{\text{guess}}}$ where P_{guess} is essentially the probability of the event when measurement of a random quantum decryption query made by $\mathcal{A}$ in the pre-challenge phase of Game $4a$ results in a ciphertext (ct, c) such that $\mathsf{ct} = \mathsf{ct}^*$ and $\mathsf{D}(\hat{\mathsf{k}}, c) \neq \perp$ for a uniformly random DEM key $\hat{\mathsf{k}}$. Note that such a ciphertext can be used to break the integrity of ciphertexts with respect to DEM, i.e., we can construct a QPT adversary $\mathcal{B}^1_{ctxt}$ such that $P_{\text{guess}} \leq \mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM},\mathcal{B}^1_{ctxt}}$ as follows.

$\mathcal{B}^1_{ctxt}$ generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and computes $(\mathsf{ct}^*, \mathsf{k}^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$. It then forwards pk to $\mathcal{A}$ and samples $i \leftarrow [q_{pre}]$. Next, $\mathcal{B}^1_{ctxt}$ proceeds to respond to $\mathcal{A}$'s quantum decryption queries in the pre-challenge phase as in Game 4a *until* the i-th query: for any ciphertext $|(\mathsf{ct}, c)\rangle$ in the computational basis, if $\mathsf{ct} = \mathsf{ct}^*$, then return $\perp$; else, compute $\mathsf{k} \leftarrow \mathsf{Decaps}(\mathsf{sk}, \mathsf{ct})$ and return $\mathsf{D}(\mathsf{k}, c)$. Next, $\mathcal{B}^1_{ctxt}$ measures the i-th decryption query, let the resulting state be $(\overline{\mathsf{ct}}, \overline{c})$. $\mathcal{B}^1_{ctxt}$ then checks if $\overline{\mathsf{ct}} = \mathsf{ct}^*$. If so, it outputs the forged DEM ciphertext $\overline{c}$ to its challenger; otherwise, $\mathcal{B}^1_{ctxt}$ aborts. It is not hard to see that $\mathcal{B}^1_{ctxt}$ wins if the event corresponding to P_{guess} in Game 4a above occurs, where $\hat{\mathsf{k}}$ is the DEM key chosen by $\mathcal{B}^1_{ctxt}$'s challenger. It follows that $P_{\text{guess}} \leq \mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM},\mathcal{B}^1_{ctxt}}$ and hence

$$|\Pr[W^{(3)}] - \Pr[W^{(4a)}]| \leq 2q_{pre}\sqrt{\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM},\mathcal{B}^1_{ctxt}}}. \qquad \square$$

Lemma 5.5. *There exists a QPT adversary $\mathcal{B}^2_{ctxt}$ such that*

$$|\Pr[W^{(4a)}] - \Pr[W^{(4b)}]| \leq 2q_{post}\sqrt{\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM},\mathcal{B}^2_{ctxt}}}.$$

Proof. We again rely on Lemma 2.1 as above to bound $|\Pr[W^{(4a)}] - \Pr[W^{(4b)}]|$. First, we simulate the pre-challenge phase of Game 4a (and 4b) as follows: let A_{pre} be an algorithm which generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$, a KEM ciphertext ct^* where $(\mathsf{ct}^*, \mathsf{k}^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$ and a uniform DEM key $\hat{\mathsf{k}}$ as generated in Game 4a above. A_{pre} then samples $b \leftarrow \{0, 1\}$ and forwards pk to the $\mathcal{A}$. It then proceeds to respond to $\mathcal{A}$'s quantum decryption queries in the pre-challenge phase as in Game 4a using sk. After receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, the algorithm A_{pre} computes $c^* \leftarrow \mathsf{E}(\hat{\mathsf{k}}, \mathsf{m}_b)$ and forwards (ct^*, c^*) to $\mathcal{A}$. At the same time, A_{pre} forwards the input $z = ((\mathsf{pk}, \mathsf{sk}), \mathsf{ct}^*, \hat{\mathsf{k}}, b)$ to the quantum oracle algorithm A_{post}. The algorithm A_{post} also has quantum access to either the post-challenge decryption oracle G in Game 4a (which uses $\hat{\mathsf{k}}$ to respond to ciphertexts of the form $|(\mathsf{ct}^*, c)\rangle$, while rejecting $|(\mathsf{ct}^*, c^*)\rangle$) or to the post-challenge decryption oracle H in Game 4b (which rejects ciphertexts of the form $|(\mathsf{ct}^*, c)\rangle$). Next, A_{post} proceeds to simulate the post-challenge phase of Game 4a or 4b towards $\mathcal{A}$ depending on whether it has access to G or H respectively by forwarding $\mathcal{A}$'s quantum decryption queries to its own oracle (i.e., G or H) and returning the corresponding output. Finally, A_{post} outputs 1 if and only if the output of $\mathcal{A}$ is b.

Observe that $\Pr[W^{(4a)}] = \Pr[1 \leftarrow A^G_{post}(z)]$ and $\Pr[W^{(4b)}] = \Pr[1 \leftarrow A^H_{post}(z)]$, and hence by Lemma 2.1 we have

$$|\Pr[W^{(4a)}] - \Pr[W^{(4b)}]| \leq 2q_{post}\sqrt{P_{\text{guess}}},$$

where P_{guess} is the probability of the event when measurement of a random quantum decryption query made by $\mathcal{A}$ in the post-challenge phase of Game $4b$ results in a (non-challenge) ciphertext (ct, c) such that $\mathsf{ct} = \mathsf{ct}^*$ and $\mathsf{D}(\hat{\mathsf{k}}, c) \neq \perp$ for a uniform DEM key $\hat{\mathsf{k}}$. We show that such a ciphertext can be used to break the integrity of ciphertexts with respect to DEM, i.e., we can construct a QPT adversary $\mathcal{B}^2_{ctxt}$ such that $P_{\text{guess}} \leq \mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}, \mathcal{B}^2_{ctxt}}$.

The description of $\mathcal{B}^2_{ctxt}$ is as follows: $\mathcal{B}^2_{ctxt}$ first generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and computes $(\mathsf{ct}^*, \mathsf{k}^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$. It forwards pk to $\mathcal{A}$ and responds to $\mathcal{A}$'s quantum decryption queries in the pre-challenge phase as in Game 4b. After receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, $\mathcal{B}^2_{ctxt}$ samples $b \leftarrow \{0, 1\}$ and forwards m_b to its (one-time) encryption oracle (provided in the INT-CTXT game of DEM). After receiving c^* from its challenger, $\mathcal{B}^2_{ctxt}$ forwards (ct^*, c^*) to $\mathcal{A}$. $\mathcal{B}^2_{ctxt}$ also samples a query number $i \leftarrow [q_{post}]$ and proceeds to respond to $\mathcal{A}$'s quantum decryption queries in the post-challenge phase as in Game 4b *until* the i-th query. Let $(\overline{\mathsf{ct}}, \overline{c})$ be the resulting state after measuring the i-th query. $\mathcal{B}^2_{ctxt}$ then checks if $\overline{\mathsf{ct}} = \mathsf{ct}^*$ and $\overline{c} \neq c^*$. If so, then it outputs $\overline{c}$; otherwise, it aborts. Note that $\mathcal{B}^2_{ctxt}$ wins if the event corresponding to P_{guess} in Game 4b above occurs. Thus, we have $P_{\text{guess}} \leq \mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}, \mathcal{B}^2_{ctxt}}$, which implies that

$$|\Pr[W^{(4a)}] - \Pr[W^{(4b)}]| \leq 2q_{post}\sqrt{\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}, \mathcal{B}^2_{ctxt}}}. \qquad \square$$

Lemma 5.6. *There is a QPT adversary $\mathcal{B}_{cpa}$ such that*

$$|\Pr[W^{(4b)}] - \frac{1}{2}| = \mathbf{Adv}^{\text{CPA}}_{\mathsf{DEM}, \mathcal{B}_{cpa}}.$$

Proof. $\mathcal{B}_{cpa}$ generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and computes $(\mathsf{ct}^*, \mathsf{k}^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$. It forwards pk to $\mathcal{A}$ and responds to $\mathcal{A}$'s quantum decryption queries in the pre-challenge phase as follows: given any $|(\mathsf{ct}, c)\rangle$ in the computational basis, if $\mathsf{ct} = \mathsf{ct}^*$, then return $\perp$; otherwise, compute $\mathsf{k} \leftarrow \mathsf{Decaps}(\mathsf{sk}, \mathsf{ct})$ and return $\mathsf{D}(\mathsf{k}, c)$. After receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, the algorithm $\mathcal{B}_{cpa}$ forwards the pair to its challenger and gets back $c^* \leftarrow \mathsf{E}(\mathsf{k}', \mathsf{m}_b)$ for a uniformly random DEM key k' and *hidden* bit b. Next, $\mathcal{B}_{cpa}$ forwards (ct^*, c^*) to $\mathcal{A}$. Then $\mathcal{B}_{cpa}$ proceeds to answer the rest of $\mathcal{A}$'s quantum decryption queries in the post-challenge phase as above while rejecting ciphertexts that are equal to (ct^*, c^*). Finally, when $\mathcal{A}$ terminates with an output $b' \in \{0, 1\}$, $\mathcal{B}_{qcca}$ terminates with the same output b'. Observe that $\mathcal{B}_{cpa}$ perfectly simulates Game 4b towards $\mathcal{A}$. It follows that

$$\mathbf{Adv}^{\text{CPA}}_{\mathsf{DEM}, \mathcal{B}_{cpa}} = \left|\Pr[W^{(4b)}] - \frac{1}{2}\right|. \qquad \square$$

By collecting all of the above bounds we get

$$\mathbf{Adv}^{\text{qCCA}}_{\mathsf{PKE}^{hyb}, \mathcal{A}} \leq \mathbf{Adv}^{\text{qCCA}}_{\mathsf{KEM}, \mathcal{B}_{qcca}} + \mathbf{Adv}^{\text{CPA}}_{\mathsf{DEM}, \mathcal{B}_{cpa}} + 2q \cdot \left(\sqrt{\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}, \mathcal{B}^1_{ctxt}}} + \sqrt{\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}, \mathcal{B}^2_{ctxt}}}\right) + \varepsilon,$$

which proves the qCCA security of PKE^{hyb}, as desired. $\square$

6 Quantum Adaptive Trapdoor Functions

In 2010, Kiltz *et al.* [KMO10] introduced the notion of *adaptive* TDFs (ATDFs) and they showed how to realize CCA-secure PKE from ATDFs. Informally, a TDF is said to be adaptive if it remains one way even if the adversary is given access to an inversion oracle. In this work, we introduce a *quantum* analog of ATDFs, namely quantum ATDFs (qATDFs), which require one-wayness to hold even if the adversary has *quantum* access to an inversion oracle. In the first part, we formally define qATDFs and prove that they imply qCCA-secure PKE. Later, we show how qATDFs can be constructed from (post-quantum) correlated-product TDFs [RS09] or lossy TDFs [PW08], which in turn can be constructed from the LWE assumption.

Definition 6.1. A trapdoor function $\mathsf{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ satisfies quantum adaptive security if for every QPT inverter $\mathcal{A}$ we have

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{qATDF}} = \Pr\left[x = x' : \begin{array}{l}(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda); x \leftarrow \{0,1\}^\lambda \\ y^* \leftarrow \mathsf{Eval}(\mathsf{ek}, x); x' \leftarrow \mathcal{A}^{|O_{y^*}(\mathsf{td},\cdot)\rangle}(\mathsf{ek}, y^*)\end{array}\right] \leq \mathrm{negl}$$

where the function $O_{y^*}(\mathsf{td}, \cdot)$ is defined as

$$O_{y^*}(\mathsf{td}, y) = \begin{cases} \perp & \text{if } y = y^*, \\ \mathsf{Invert}(\mathsf{td}, y) & \text{otherwise.} \end{cases}$$

(Similar to Definition 2.2, we also encode $\perp$ to be a bitstring outside $\{0,1\}^\lambda$ in order to properly define the result $z \oplus \perp$ in the output register of $|O_{y^*}(\mathsf{td}, \cdot)\rangle$ described above.)

Note that adaptive one-wayness for TDFs defined in [KMO10] differs from Definition 6.1 only in that $\mathcal{A}$ has *classical* access to the oracle $O_{y^*}(\mathsf{td}, \cdot)$. It is not hard to extend the separation result in [BZ13b, Subsection 4.1] to TDFs, which implies that our notion of qATDFs is *strictly* stronger than ATDFs.

6.1 Quantum CCA Security from Quantum ATDFs

Kiltz *et al.* [KMO10] showed a construction of *classically* CCA-secure PKE from any ATDF. We prove that the *same* construction results in a qCCA-secure PKE if the underlying ATDF satisfies quantum security. To be more specific, [KMO10] constructs a *single-bit* CCA-secure PKE from an ATDF and then relies on the "single-bit to multi-bit" compiler of [Ms09, HLW12]. In the quantum setting, we follow the same blueprint to first build a single-bit qCCA secure PKE from a qATDF, and then we show that the "single-bit to multi-bit" compiler of [HLW12] also extends to qCCA-secure PKE.

Let $\mathsf{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ be a TDF and $\mathrm{GL}(\cdot)$ be the corresponding Goldreich-Levin hardcore bit [GL89].[1] We construct a single-bit PKE scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

$\mathsf{Gen}(1^\lambda)$: Run $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$, and set $(\mathsf{pk}, \mathsf{sk}) := (\mathsf{ek}, \mathsf{td})$. Return $(\mathsf{pk}, \mathsf{sk})$.

```
Enc(pk, m): For $i = 1, \ldots, \lambda$, do:
                $x \leftarrow \{0,1\}^\lambda$; $h \leftarrow \mathrm{GL}(x)$; if $h = \mathsf{m}$, return $\mathsf{Eval}(\mathsf{pk}, x)||0$.
            Return $\mathsf{m}||1$.
```

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: Parse $\mathsf{ct} \to (\mathsf{ct}_1||b)$ with $b \in \{0,1\}$. If $b = 1$, return ct_1; else return $\mathrm{GL}(\mathsf{Invert}(\mathsf{sk}, \mathsf{ct}_1))$.

Theorem 6.2. *If* TDF *is a qATDF, then* PKE *is a (single-bit) qCCA secure PKE.*

[1] It is not hard to see that the Goldreich-Levin theorem relating the one-wayness of a TDF to the hardcore bit security also applies when the TDF inverter and the bit distinguisher have *quantum* access to the corresponding TDF inversion oracle. This is because the probability-theoretic analysis in the original Goldreich-Levin theorem [GL89] is agnostic of any oracle access (be it classical or quantum) that the inverter and distinguisher have; the oracles would only be needed to ensure that the inverter can properly simulate the distinguisher's view.

Proof. Let $\mathcal{A}$ be any QPT adversary that breaks the qCCA security of PKE while making q quantum decryption queries, where q_{pre}/q_{post} denotes the number of queries in the pre/post-challenge phase. Consider the following games:

Game 1: This is essentially the same as the qCCA game for PKE, except for some changes in the setup.[1]

- Generate $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$, and set $(\mathsf{pk}, \mathsf{sk}) := (\mathsf{ek}, \mathsf{td})$. Then for all $i \in [\lambda]$, generate tuples $\big((y_i, h_i)\big)_{i\in[\lambda]}$ where $y_i = \mathsf{Eval}(\mathsf{pk}, x_i)$ for uniformly random $x_i \leftarrow \{0,1\}^\lambda$ and $h_i = \mathrm{GL}(x_i)$. Also sample a random bit $b \leftarrow \{0, 1\}$.
- Forward pk to the adversary $\mathcal{A}$ and respond to $\mathcal{A}$'s quantum decryption queries in the normal way using the description of $\mathsf{Dec}(\mathsf{sk}, \cdot)$ above.
- After receiving a pair of messages $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, find the least $i^* \in [\lambda]$ such that $h_{i^*} = \mathsf{m}_b$. If no such i^* exists, compute the challenge ciphertext ct^* as $\mathsf{ct}^* := \mathsf{m}_b \parallel 1$; otherwise, set $\mathsf{ct}^* := y_{i^*} \parallel 0$.
- Again respond to $\mathcal{A}$'s quantum decryption queries in the normal way as above, but this time making sure to reject ciphertexts that are equal to ct^*.
- $\mathcal{A}$ then terminates with an output $b' \in \{0, 1\}$.

Game 2a: In this game, we modify the decryption oracle *post-challenge* phase such that it rejects ciphertexts (i.e., returns $\perp$) that are equal to $(y_i \parallel 0)$ for some $i \in [\lambda]$, in addition to rejecting ciphertexts equal to ct^*.[2]

Game 2b: In this game, we modify the decryption oracle *pre-challenge* phase such that it also rejects ciphertexts that are equal to $(y_i \parallel 0)$ for some $i \in [\lambda]$.

Game 3: In this game, we sample $h_i \leftarrow \{0, 1\}$, instead of $h_i = \mathrm{GL}(x_i)$.

Game 4: During the computation of challenge ciphertext ct^*, when no $i^* \in [\lambda]$ satisfying $h_{i^*} = \mathsf{m}_b$ exists, we set $\mathsf{ct}^* := \perp$ (instead of $\mathsf{ct}^* := \mathsf{m}_b \parallel 1$).

Now we define $W^{(j)}$, for $j \in \{1, 2a, 2b, 3, 4\}$, to be the event when $\mathcal{A}$ succeeds in guessing the bit b (i.e., $b' = b$) in Game j. By definition, we have

$$\mathbf{Adv}^{\mathsf{qCCA}}_{\mathsf{PKE},\mathcal{A}} = \Big|\Pr[W^{(1)}] - \frac{1}{2}\Big|.$$

Quantum CCA security of the scheme PKE follows from the following lemmas.

Lemma 6.3. $|\Pr[W^{(1)}] - \Pr[W^{(2a)}]| \leq 2q_{post}\sqrt{\frac{\lambda}{2^{\lambda-1}}}$.

Proof. Here we use Lemma 2.1 to bound $|\Pr[W^{(1)}] - \Pr[W^{(2a)}]|$. We simulate the pre-challenge phase of Game 1 (and 2a) using a QPT algorithm A_{pre} which first receives as input a pair $(\mathsf{pk}, \mathsf{sk})$ and $((y_i, h_i))_{i\in[\lambda]}$ as generated in Game 1 (and 2a) above. A_{pre} proceeds to simulate the pre-challenge phase (using sk to respond to $\mathcal{A}$'s quantum decryption queries). After computing ct^*, (by applying Lemma 2.1) A_{pre} forwards the input $z = (((\mathsf{pk}, \mathsf{sk}), ((y_i, h_i))_{i\in[\lambda]}), \mathsf{ct}^*, b)$ to the oracle algorithm A_{post}. The algorithm A_{post} has quantum access either to the corresponding post-challenge decryption oracle $G := \mathsf{Dec}(\mathsf{sk}, \cdot)$ in Game 1 (which also rejects ciphertexts equal to ct^*) or to the (modified) decryption oracle H in Game 2a (which additionally rejects non-challenge ciphertexts of the form $|c\rangle \in \{|y_i \parallel 0\rangle \mid i \in [\lambda]\}$). Note that the outputs of oracles G and H differ with respect to the set $S = \{\mathsf{ct} = (y_i \parallel 0) \mid i \in [\lambda] \wedge \mathsf{ct} \neq \mathsf{ct}^*\}$. Next, A_{post} proceeds to simulate the post-challenge phase of Game 1 or Game 2a towards $\mathcal{A}$ depending on whether it has access to G or H respectively by forwarding $\mathcal{A}$'s quantum decryption queries to its own oracle (i.e., G or H) and returning the corresponding output. Finally, A_{post} outputs 1 if the output of $\mathcal{A}$ is b.

Observe that $\Pr[W^{(1)}] = \Pr[1 \leftarrow A^G_{post}(z)]$ and $\Pr[W^{(2a)}] = \Pr[1 \leftarrow A^H_{post}(z)]$. Thus we have $|\Pr[W^{(1)}] - \Pr[W^{(2a)}]| \leq 2q_{post}\sqrt{P_{\text{guess}}}$ where P_{guess} denotes the the probability of the event when measurement of a random

[1] We "pre-compute" the randomness $(x_i)_{i\in[\lambda]}$ (used to encrypt $\mathcal{A}$'s chosen messages in the challenge phase) already in the *pre-challenge* phase. But this does not affect $\mathcal{A}$'s view in any way compared to the original qCCA game.

[2] In contrast to qCCA security proofs in earlier sections (e.g., Section 3) here we are first modifying the decryption oracle in the post-challenge phase followed by the pre-challenge phase. This step is crucial in our analysis as will be seen later on.

quantum decryption query made by $\mathcal{A}$ in the post-challenge phase of Game 1 results in a non-challenge ciphertext $\mathsf{ct} := (\mathsf{ct}_1 \parallel b) \in S$ ($b \in \{0,1\}$). To bound $P_{\text{guess}} := \Pr[(\mathsf{ct}_1 \parallel b) \in S]$, we consider $b = 0$ (if $b = 1$, the corresponding probability is zero). Now we have

$$\Pr[(\mathsf{ct}_1 \parallel 0) \in S] \leq \Pr[\exists i \in [\lambda] \text{ s.t. } \mathsf{ct}_1 = y_i \wedge \mathsf{ct} \neq \mathsf{ct}^*] \leq \Pr[\exists i \in [\lambda] \text{ s.t. } \mathsf{ct}_1 = y_i \mid \mathsf{ct} \neq \mathsf{ct}^*].$$

Consider the case when $\mathsf{ct}^* = y_{i^*} \parallel 0$ for some $i^* \in [\lambda]$. This means that for $i < i^*$, we have $\mathrm{GL}(x_i) = 1 - \mathsf{m}_b$. Here we use a *statistical* fact that conditioning on $\mathrm{GL}(x_i) = 1 - \mathsf{m}_b$ reduces the min-entropy of x_i by one bit. Hence, $\mathcal{A}$'s view in the post-challenge phase of Game 1 is independent of the values $((y_i, h_i))_{i^* < i \leq \lambda}$ (as well as the corresponding x_i's) and *conditionally* independent of the values $((y_i, h_i))_{1 \leq i < i^*}$, conditioned on $\mathrm{GL}(x_i) = 1 - \mathsf{m}_b$ (because the decryption oracle in Game 1 *does not* use the y_i values to reject ciphertexts yet). Therefore, to analyze $\Pr[\exists i \in [\lambda] \text{ s.t. } \mathsf{ct}_1 = y_i \mid \mathsf{ct}_1 \neq y_{i^*}]$, it is easier to consider the values $((y_i, h_i))_{i \in [\lambda]}$ being generated (with an appropriate distribution following the conditional independence noted above) *after* measuring $\mathcal{A}$'s decryption query to ct. Observe that after measurement, each x_i for $1 \leq i < i^*$ (respectively, for $i^* < i \leq \lambda$) is sampled independently from an entropy source of $\lambda - 1$ bits (respectively, λ bits). Moreover, since $\mathsf{Eval}(\mathsf{pk}, \cdot)$ is an injection, the probability that any y_i ($i \in [\lambda]$) coincides with the measured ct_1 is at most $1/2^{\lambda-1}$. Hence by applying a union bound, we get

$$\Pr[\exists i \in [\lambda] \text{ s.t. } \mathsf{ct}_1 = y_i \mid \mathsf{ct}_1 \neq y_{i^*}] \leq (\lambda - 1) \cdot \frac{1}{2^{\lambda-1}}.$$

Similarly, if $\mathsf{ct}^* = \mathsf{m}_b \parallel 1$ we have $\mathrm{GL}(x_i) = 1 - \mathsf{m}_b$ for all $i \in [\lambda]$. Hence, $\mathcal{A}$'s view in the post-challenge phase of Game 1 is *conditionally* independent of the values $((y_i, h_i))_{i \in [\lambda]}$, conditioned on $\mathrm{GL}(x_i) = 1 - \mathsf{m}_b$. This time, after measuring $\mathcal{A}$'s random decryption query to ct, we have each x_i ($i \in [\lambda]$) to be sampled independently from an entropy source of $\lambda - 1$ bits. Therefore, by a similar analysis as above, it follows that

$$\Pr[\exists i \in [\lambda] \text{ s.t. } \mathsf{ct}_1 = y_i \mid \mathsf{ct} \neq (\mathsf{m}_b \parallel 1)] \leq \lambda \cdot \frac{1}{2^{\lambda-1}}.$$

By an averaging argument we have $P_{\text{guess}} \leq \lambda \cdot \frac{1}{2^{\lambda-1}}$, as required. □

Lemma 6.4. $|\Pr[W^{(2a)}] - \Pr[W^{(2b)}]| \leq 2q_{pre}\sqrt{\frac{\lambda}{2^\lambda}}$.

Proof. We use Lemma 2.1 as to bound $|\Pr[W^{(2a)}] - \Pr[W^{(2b)}]|$. Let A be a quantum oracle algorithm which receives as input a pair $(\mathsf{pk}, \mathsf{sk})$ and $((y_i, h_i))_{i \in [\lambda]}$ as generated in Game 2a (and 2b) above; $z = ((\mathsf{pk}, \mathsf{sk}), \langle (y_i, h_i) \rangle_{i \in [\lambda]})$. A also has quantum access to either the corresponding pre-challenge decryption oracle $G := \mathsf{Dec}(\mathsf{sk}, \cdot)$ in Game 2a or to the oracle H in Game 2b (which rejects ciphertexts $|c\rangle \in \{|y_i \parallel 0\rangle \mid i \in [\lambda]\}$). Note that the outputs of oracles G and H differ with respect to the set $S = \{\mathsf{ct} = (y_i \parallel 0) \mid i \in [\lambda]\}$. Using its input z, A proceeds to simulate either Game 2a or 2b towards $\mathcal{A}$ depending on whether it has access to G or H, respectively. It is worth mentioning that A responds to $\mathcal{A}$'s quantum decryption queries in the pre-challenge phase using its own oracle (i.e., G or H), while in the post-challenge phase, A uses the secret key sk for quantum decryption while rejecting ciphertexts equal to $(y_i \parallel 0)$ for some $i \in [\lambda]$ or equal to ct^*. Finally A outputs 1 if and only if the output of $\mathcal{A}$ equals b.

Similar to the previous case, we have $|\Pr[W^{(2a)}] - \Pr[W^{(2b)}]| \leq 2q_{pre}\sqrt{P_{\text{guess}}}$ where P_{guess} is essentially the probability of the event when measurement of a random quantum decryption query made by $\mathcal{A}$ in the pre-challenge phase of Game 2a results in a ciphertext $(\mathsf{ct}_1 \parallel b) \in S$ ($b \in \{0,1\}$). To bound P_{guess}, note that $\mathcal{A}$'s view in the pre-challenge phase of Game 2a is completely independent of the values $((y_i, h_i))_{i \in [\lambda]}$ (as well as the corresponding x_i) because these values are *only* used starting from the challenge phase. Hence to analyze the term $\Pr[(\mathsf{ct}_1 \parallel b) \in S]$, it is easier to consider the values $((y_i, h_i))_{i \in [\lambda]}$ being generated *after* measuring $\mathcal{A}$'s decryption query to $(\mathsf{ct}_1 \parallel b)$. In the rest of this analysis, we consider $b = 0$ (if $b = 1$, the corresponding probability is zero). After measurement, each x_i ($i \in [\lambda]$) is sampled independently from an entropy source of λ bits. Since $\mathsf{Eval}(\mathsf{pk}, \cdot)$ is an injection, the probability that any y_i coincides with the measured ct_1 is $1/2^\lambda$. By applying a union bound, we get $P_{\text{guess}} \leq \lambda \cdot 2^{-\lambda}$. □

Remark 6.5. If we modify the decryption oracle in the pre-challenge phase first followed by the post-challenge phase, then we cannot use a similar argument as in the proof of Lemma 6.3 to claim that $\mathcal{A}$'s view in the post-challenge phase is (conditionally) independent of the values $((y_i, h_i))_{i\in[\lambda]}$ (and that we can generate the values $((y_i, h_i))_{i\in[\lambda]}$ *after* measuring $\mathcal{A}$'s decryption query in post-challenge phase). To see this, note that because of our prior modification to the decryption oracle in pre-challenge phase, the values $((y_i, h_i))_{i\in[\lambda]}$ are implicitly generated to be used to reject ciphertexts, and hence $\mathcal{A}$'s view in the pre-challenge (and post-challenge) phase would then depend on these values.

Lemma 6.6. *There exists a QPT distinguisher $\mathcal{D}$ such that* $|\Pr[W^{(2b)}] - \Pr[W^{(3)}]| = 2 \cdot \mathbf{Adv}^{\text{hcDist}}_{GL,\mathcal{D}}$.

Proof. The description of $\mathcal{D}$ is as follows. It gets as input $(\mathsf{ek}, ((y_i, h_i))_{i\in[\lambda]})$ where $y_i = \mathsf{Eval}(\mathsf{ek}, x_i)$ for $x_i \leftarrow \{0,1\}^\lambda$, and either $h_i = \mathrm{GL}(x_i)$ or $h_i \leftarrow \{0,1\}$. D also has quantum access to an oracle O implementing the function $\mathsf{Invert}(\mathsf{td}, \cdot)$ while rejecting inputs equal to y_i (i.e., returns $\perp$) for some $i \in [\lambda]$. $\mathcal{D}$ then proceeds to simulate either Game 2b or Game 3 by forwarding ek to $\mathcal{A}$ and responding to $\mathcal{A}$'s quantum decryption queries as in Game 2b (and 3) using the oracle O. $\mathcal{D}$ also uses $((y_i, h_i))_{i\in[\lambda]})$ to compute the challenge ciphertext ct^* (after first sampling a bit $b \leftarrow \{0,1\}$). Finally, when $\mathcal{A}$ terminates with a bit b', $\mathcal{D}$ outputs 1 if $b = b'$ and outputs 0 otherwise. The proof is complete by observe that

$$\Pr[1 \leftarrow \mathcal{D}(\mathsf{ek}, ((y_i, h_i))_{i\in[\lambda]})|h_i = \mathrm{GL}(x_i)] = \Pr[W^{(2b)}]$$

and

$$\Pr[1 \leftarrow \mathcal{D}(\mathsf{ek}, ((y_i, h_i))_{i\in[\lambda]})|h_i \leftarrow \{0,1\}] = \Pr[W^{(3)}].$$

□

Lemma 6.7. $|\Pr[W^{(3)}] - \Pr[W^{(4)}]| \le 2^{-\lambda}$.

Proof. Note that Games 3 and 4 proceed identically unless there does not exist an $i^* \in [\lambda]$ such that $h_{i^*} = \mathsf{m}_b$, which happens with probability $2^{-\lambda}$. □

Lemma 6.8. $\Pr[W^{(4)}] = 1/2$.

Proof. In Game 4, note that $\mathcal{A}$'s view is completely independent of the bit b. □

6.2 Quantum ATDFs from Correlated-Product TDFs

Here we show that the ATDF construction of Kiltz *et al.* [KMO10] from correlated-product TDFs (CP-TDFs) satisfies *quantum* security if the underlying CP-TDF is post-quantum secure, i.e., we prove that (post-quantum) CP-TDFs are sufficient to realize quantum ATDFs. We recall the definition of CP-TDFs from [RS09].

Definition 6.9. A trapdoor function $\mathsf{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ is t-correlated-product (t-CP-TDF) one-way if for every QPT inverter $\mathcal{A}$, it holds that

$$\mathbf{Adv}^{\mathsf{CPOW}}_{\mathsf{TDF},\mathcal{A}} = \Pr[x' = x : x' \leftarrow \mathcal{A}((\mathsf{ek}_i)_{i\in[t]}, \mathbf{y}^*)] \le \mathrm{negl},$$

where $(\mathsf{ek}_i, \mathsf{td}_i) \leftarrow \mathsf{Gen}(1^\lambda)$ for $i \in [t]$, $x \leftarrow \{0,1\}^\lambda$, and $\mathbf{y}^* = (\mathsf{Eval}(\mathsf{ek}_i, x))_{i\in[t]}$.

Let $\mathsf{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ be a TDF with fixed output length $n = n(\lambda)$. Now we recall the construction of ATDF $\overline{\mathsf{TDF}} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Eval}}, \overline{\mathsf{Invert}})$ from TDF in [KMO10] as follows.

$\overline{\mathsf{Gen}}(1^\lambda)$: Let $(\mathsf{ek}_0, \mathsf{td}_0) \leftarrow \mathsf{Gen}(1^\lambda)$. Sample $(\mathsf{ek}_i^b, \mathsf{td}_i^b) \leftarrow \mathsf{Gen}(1^\lambda)$ for $i \in [n]$ and $b \in \{0,1\}$. Output $(\mathsf{ek}, \mathsf{td})$ where

$$\mathsf{ek} := (\mathsf{ek}_0, ((\mathsf{ek}_i^0, \mathsf{ek}_i^1))_{i\in[n]}), \quad \mathsf{td} := (\mathsf{td}_0, ((\mathsf{td}_i^0, \mathsf{td}_i^1))_{i\in[n]}).$$

$\overline{\mathsf{Eval}}(\mathsf{ek}, x)$: Output $(\mathsf{Eval}(\mathsf{ek}_0, x) \parallel \mathsf{Eval}(\mathsf{ek}_1^{\mathsf{b}_1}, x) \parallel \ldots \parallel \mathsf{Eval}(\mathsf{ek}_n^{\mathsf{b}_n}, x))$, where b_i denotes the i-th bit of $\mathsf{b} := \mathsf{Eval}(\mathsf{ek}_0, x)$ for $i \in [n]$.

$\overline{\mathsf{Invert}}(\mathsf{td}, y)$: Parse $y \to (\mathsf{b} \parallel y_1 \parallel \ldots \parallel y_n)$. Let $x \leftarrow \mathsf{Invert}(\mathsf{td}_0, \mathsf{b})$. Return x if $x = \mathsf{Invert}(\mathsf{td}_i^{\mathsf{b}_i}, y_i)$ for all $i \in [n]$. Otherwise, return $\perp$.

Theorem 6.10. *If* TDF *is an* $(n+1)$*-CP-TDF, then* $\overline{\mathsf{TDF}}$ *is a quantum ATDF.*

Proof. The proof is quite similar to that of [KMO10, Theorem 3]. Given any efficient inverter $\mathcal{A}$ breaking the adaptive one-wayness of $\overline{\mathsf{TDF}}$, they describe an efficient inverter $\mathcal{B}$ breaking the one-way $(n+1)$-correlated-product of TDF. In their reduction, $\mathcal{B}$ simulates the adaptive one-wayness game with respect to $\overline{\mathsf{TDF}}$ towards $\mathcal{A}$ since it is able to (classically) implement the $\overline{\mathsf{TDF}}$ inversion functionality using some trapdoor information. It is easy to see that their reduction can be extended to the *quantum* setting since $\mathcal{B}$ can also simulate this $\overline{\mathsf{TDF}}$ inversion oracle in quantum superposition using the same trapdoor information (we are using the fact, mentioned in Section 2, that any function which has an efficient classical algorithm computing it can be implemented efficiently as a quantum-accessible oracle).□

Quantum ATDF from LWE. We briefly describe two ways to instantiate quantum ATDFs from the LWE assumption. For the first approach, recall that Rosen and Segev [RS09] showed lossy TDFs imply TDFs with correlated security. By using the LWE-based lossy TDF construction of [PW08] and relying on our construction of quantum ATDFs from TDFs with correlated security, we get an instantiation of quantum ATDFs from the LWE assumption. An alternative (and more efficient) approach is to rely on the LWE-based TDF construction of [MP13] (which also satisfies correlated security), which in turn yields a construction of quantum ATDF from LWE (based on our generic transformation).

7 Completeness of Bit Encryption for Quantum CCA Security

Given a single-bit CCA-secure PKE, [Ms09] (and later [HLW12]) showed how to realize multi-bit CCA-secure PKE. In this section, we show an analogous result with respect to *quantum* CCA security by extending the framework of [HLW12] to the quantum setting. To obtain a generic single-bit to multi-bit compiler, Hohenberger *et al.* [HLW12] introduced a new cryptographic primitive called *Detectable Chosen Ciphertext Secure* (DCCA-secure) PKE. In the first step, they showed a construction of (multi-bit) CCA-secure PKE scheme from any (multi-bit) DCCA-secure PKE. In the second step, they argue that single-bit CCA-secure PKE can be used to construct multi-bit DCCA-secure PKE.

Following a similar approach, we first define a quantum analog of DCCA security called *quantum* DCCA (qDCCA) security, and we prove that (multi-bit) qCCA-secure PKE is implied by (multi-bit) qDCCA security. Next, we show that single-bit qCCA-secure PKE is sufficient to realize multi-bit qDCCA-secure PKE, thereby proving the completeness of bit encryption for qCCA security.

First, we recall the notion of *detectable* PKE from [HLW12]. A detectable PKE $\mathsf{DPKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, F)$ is a tuple of PPT algorithms where $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ follow the same definition as in a normal PKE scheme, and the *detecting* function F (which is efficiently computable) takes as input a public key pk and two ciphertexts $\mathsf{ct}, \mathsf{ct}'$, and outputs a bit. Correctness of a detectable PKE scheme is same as a normal PKE. We now formally define the qDCCA security property of such schemes.

Definition 7.1. A detectable PKE scheme $\mathsf{DPKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, F)$ is said to be qDCCA secure if the following two properties hold:

- Quantum unpredicability of F: For every QPT adversary $\mathcal{A}$, it holds that

$$\mathbf{Adv}^{\text{qPredict}}_{\mathsf{DPKE},\mathcal{A}} = \Pr\left[F(\mathsf{pk}, \mathsf{ct}^*, \mathsf{ct}) = 1 : \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) \\ (\mathsf{m}, \mathsf{ct}) \leftarrow \mathcal{A}^{|\mathsf{Dec}(\mathsf{sk},\cdot)\rangle}(\mathsf{pk}) \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \end{array} \right] \leq \text{negl}\,.$$

- Quantum indistinguishability of encryptions: For every QPT adversary $\mathcal{B}$ we have

$$\mathbf{Adv}^{\text{qInd}}_{\mathsf{DPKE},\mathcal{B}} = \left| \Pr\left[b = b' : \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda); b \leftarrow \{0,1\} \\ (\mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \leftarrow \mathcal{B}^{|\mathsf{Dec}(\mathsf{sk},\cdot)\rangle}(\mathsf{pk}) \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b); b' \leftarrow \mathcal{B}^{|O^F_{\mathsf{ct}^*}(\mathsf{sk},\cdot)\rangle}(\mathsf{ct}^*, \mathsf{st}) \end{array} \right] - \frac{1}{2} \right| \leq \text{negl}$$

where the function $O^F_{\mathsf{ct}^*}(\mathsf{sk}, \cdot)$ is defined as

$$O^F_{\mathsf{ct}^*}(\mathsf{sk}, \mathsf{ct}) = \begin{cases} \bot & \text{if } F(\mathsf{pk}, \mathsf{ct}^*, \mathsf{ct}) = 1 \text{ or } \mathsf{ct} = \mathsf{ct}^*, \\ \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) & \text{otherwise.} \end{cases}$$

In the definition above st is some arbitrary state information. We require the messages m_0 and m_1 to be of the same length. We also encode $\bot$ to be a bitstring outside the message space of DPKE in order to properly define $z \oplus \bot$ in the output register of $|O^F_{\mathsf{ct}^*}(\mathsf{sk}, \cdot)\rangle$ described above.

The notion of DCCA security defined in [HLW12] differs from Definition 7.1 in that the adversary $\mathcal{A}$ (or $\mathcal{B}$ in case of indistinguishability) above has *classical* access to its corresponding oracle(s).

7.1 qCCA-Secure PKE from qDCCA-Secure PKE

We show how to realize qCCA-secure PKE from qDCCA-secure detectable PKE. Specifically, we require the following three building blocks: a *well-spread* 1-bounded CCA secure PKE[1] $\mathsf{PKE}_{1-cca} = (\mathsf{Gen}_{1-cca}, \mathsf{Enc}_{1-cca}, \mathsf{Dec}_{1-cca})$, a CPA-secure PKE $\mathsf{PKE}_{cpa} = (\mathsf{Gen}_{cpa}, \mathsf{Enc}_{cpa}, \mathsf{Dec}_{cpa})$, and also a qDCCA-secure detectable PKE $\mathsf{DPKE} = (\mathsf{Gen}_{qdcca}, \mathsf{Enc}_{qdcca}, \mathsf{Dec}_{qdcca}, F)$ which are perfectly correct.[2] Consider the following scheme $\overline{\mathsf{PKE}} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$:

$\overline{\mathsf{Gen}}(1^\lambda)$: Run $(\mathsf{pk}_{in}, \mathsf{sk}_{in}) \leftarrow \mathsf{Gen}_{qdcca}(1^\lambda)$, $(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{Gen}_{1-cca}(1^\lambda)$ and $(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{Gen}_{cpa}(1^\lambda)$. Output

$$\mathsf{pk} = (\mathsf{pk}_{in}, \mathsf{pk}_A, \mathsf{pk}_B), \quad \mathsf{sk} := (\mathsf{sk}_{in}, \mathsf{sk}_A, \mathsf{sk}_B).$$

$\overline{\mathsf{Enc}}(\mathsf{pk}, \mathsf{m})$: Sample $r_{in}, r_A, r_B \leftarrow \{0,1\}^\lambda$ and output $\mathsf{ct} = (\mathsf{ct}_A, \mathsf{ct}_B)$ where

$$\mathsf{ct}_{in} = \mathsf{Enc}_{qdcca}(\mathsf{pk}_{in}, (r_A, r_B, \mathsf{m}); r_{in}),\ \ \mathsf{ct}_A = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}_{in}; r_A),\ \ \mathsf{ct}_B = \mathsf{Enc}_{cpa}(\mathsf{pk}_B, \mathsf{ct}_{in}; r_B).$$

$\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct} = (\mathsf{ct}_A, \mathsf{ct}_B))$: Compute the following

$$\mathsf{ct}_{in} = \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A), \quad (r_A, r_B, \mathsf{m}) = \mathsf{Dec}_{qdcca}(\mathsf{sk}_{in}, \mathsf{ct}_{in}).$$

If $\mathsf{ct}_A = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}_{in}; r_A)$ and $\mathsf{ct}_B = \mathsf{Enc}_{cpa}(\mathsf{pk}_B, \mathsf{ct}_{in}; r_B)$, return m; otherwise, return $\bot$.

Theorem 7.2. *If PKE_{1-cca} is 1-bounded CCA secure and well-spread, PKE_{cpa} is CPA secure, and DPKE is qDCCA secure, then $\overline{\mathsf{PKE}}$ is qCCA secure.*

Proof. As in the classical CCA security proof of [HLW12], we consider a variant of the qCCA security game (specific to $\overline{\mathsf{PKE}}$) where the challenger either encrypts one of two challenge messages or encrypts a string of zeros only when computing the "inner" ciphertext ct^*_{in}. We describe this *nested quantum indistinguishability* game with respect to a QPT adversary $\mathcal{A}$ as follows:

- Sample three pairs as $(\mathsf{pk}_{in}, \mathsf{sk}_{in}) \leftarrow \mathsf{Gen}_{qdcca}(1^\lambda)$, $(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{Gen}_{1-cca}(1^\lambda)$, and $(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{Gen}_{cpa}(1^\lambda)$. Set $\mathsf{pk} = (\mathsf{pk}_{in}, \mathsf{pk}_A, \mathsf{pk}_B)$, $\mathsf{sk} = (\mathsf{sk}_{in}, \mathsf{sk}_A, \mathsf{sk}_B)$. Sample random bits $b, z \leftarrow \{0,1\}$.
- Forward pk to $\mathcal{A}$. Next, $\mathcal{A}$ has *quantum* access to the decryption oracle $\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)$. Afterwards, $\mathcal{A}$ forwards $(\mathsf{m}_0, \mathsf{m}_1)$.
- Sample $r_A, r_B \leftarrow \{0,1\}^\lambda$. If $z = 0$, compute $\mathsf{ct}^*_{in} \leftarrow \mathsf{Enc}_{qdcca}(\mathsf{pk}_{in}, (r_A, r_B, \mathsf{m}_b))$. Otherwise, let $\mathsf{ct}^*_{in} \leftarrow \mathsf{Enc}_{qdcca}(\mathsf{pk}_{in}, 0^\ell)$.[3] Send $\mathsf{ct}^* = (\mathsf{ct}^*_A, \mathsf{ct}^*_B)$ to $\mathcal{A}$ where

$$\mathsf{ct}^*_A = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}^*_{in}; r_A), \quad \mathsf{ct}^*_B = \mathsf{Enc}_{cpa}(\mathsf{pk}_B, \mathsf{ct}^*_{in}; r_B).$$

[1] 1-bounded CCA-secure PKE can be built from any CPA-secure PKE [CHH+07]. As observed in [FO13], any CPA-secure PKE can be made well-spread (Section 2) by appending independent random strings to the end of ciphertexts, and the CPA to 1-bounded CCA transformation of [CHH+07] preserves well-spreadness.

[2] We make this assumption of perfect correctness for ease of exposition. One can extend our following qCCA security proof to the case when the underlying PKE schemes are almost-all-keys-correct, similar to our qCCA security analysis in Section 4.

[3] ℓ denotes the bit-length of (r_A, r_B, m_b).

- $\mathcal{A}$ has "post-challenge" quantum access to the oracle $\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)$, which rejects ciphertexts equal to ct^*.
- Finally, $\mathcal{A}$ outputs a bit z' and is said to win this game if $z' = z$.

We define the advantage of adversary $\mathcal{A}$ with respect to the $\overline{\mathsf{PKE}}$ construction as

$$\mathbf{Adv}^{\text{Nest-qInd}}_{(\mathsf{PKE}_{1-cca}, \mathsf{PKE}_{cpa}, \mathsf{DPKE}), \mathcal{A}} = |\Pr[z' = z] - 1/2|,$$

and the construction is said to be Nest-qCCA secure if the quantity $\mathbf{Adv}^{\text{Nest-qInd}}_{(\mathsf{PKE}_{1-cca}, \mathsf{PKE}_{cpa}, \mathsf{DPKE}), \mathcal{A}}$ is negligible.

It is easy to see that if $\overline{\mathsf{PKE}}$ is Nest-qCCA secure, then it is also qCCA-secure. Thus, towards proving the qCCA security of $\overline{\mathsf{PKE}}$, we will instead be focusing on its Nest-qCCA security. Let $\mathcal{A}$ be any QPT adversary that breaks the Nest-qCCA security of $\overline{\mathsf{PKE}}$ while making at most q quantum decryption queries (with at most q_{post} queries in the "post-challenge" phase). Also, let PKE_{1-cca} be γ-spread. Consider the following sequence of games:

- Game 1: This is the nested quantum indistinguishability game for $\overline{\mathsf{PKE}}$.
- Game 2: In this game, we modify the decryption oracle *post-challenge phase* such that it additionally rejects ciphertexts $|\mathsf{ct}\rangle = |(\mathsf{ct}_A, \mathsf{ct}_B)\rangle$ when $\mathsf{ct}_A = \mathsf{ct}_A^*$.
- Game 3: In this game, we modify the decryption oracle in the post-challenge phase such that it also rejects ciphertexts $|\mathsf{ct}\rangle = |(\mathsf{ct}_A, \mathsf{ct}_B)\rangle$ for which we have $F(\mathsf{pk}_{in}, \mathsf{ct}_{in}^*, \mathsf{ct}_{in}) = 1$ where $\mathsf{ct}_{in} = \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A)$.
- Game 4: In this game, during the computation of ct^*, we compute ct_B^* as $\mathsf{ct}_B^* = \mathsf{Enc}_{cpa}(\mathsf{pk}_B, 1^k; r_B)$ where k is the bit-length of ct_{in}^*.
- Game 5: In this game, during the computation of ct^*, we compute ct_A^* as $\mathsf{ct}_A^* = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, 1^k; r_A)$ where k is the bit-length of ct_{in}^*.
- Game 6: We revert the changes to the decryption oracle post-challenge phase in Game 3. Now the decryption oracle additionally *only* rejects ciphertexts $|\mathsf{ct}\rangle = |(\mathsf{ct}_A, \mathsf{ct}_B)\rangle$ where $\mathsf{ct}_A = \mathsf{ct}_A^*$. Otherwise, the oracle implements $\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)$.

Now we define $W^{(j)}$, for $j \in [6]$, to be the event when $\mathcal{A}$ succeeds in guessing the bit z (i.e., $z' = z$) in Game j. By definition, we have

$$\mathbf{Adv}^{\text{Nest-qInd}}_{(\mathsf{PKE}_{1-cca}, \mathsf{PKE}_{cpa}, \mathsf{DPKE}), \mathcal{A}} = \left| \Pr[W^{(1)}] - \frac{1}{2} \right|.$$

Lemma 7.3. $|\Pr[W^{(1)}] - \Pr[W^{(2)}]| \leq 2^{-\gamma}$.

Proof. Observe that Games 1 and 2 proceed identically unless there exists a $\mathsf{ct} = (\mathsf{ct}_A, \mathsf{ct}_B)$ such that $\mathsf{ct}_A = \mathsf{ct}_A^*$, $\mathsf{ct}_B \neq \mathsf{ct}_B^*$, and $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) \neq \bot$. If $z = 0$, we have $\mathsf{ct}_{in}^* \leftarrow \mathsf{Enc}_{qdcca}(\mathsf{pk}_{in}, (r_A, r_B, \mathsf{m}_b))$, $\mathsf{ct}_A^* = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}_{in}^*; r_A)$ and $\mathsf{ct}_B^* = \mathsf{Enc}_{cpa}(\mathsf{pk}_B, \mathsf{ct}_{in}^*; r_B)$. For any ciphertext ct with $\mathsf{ct}_A = \mathsf{ct}_A^*$ and $\mathsf{ct}_B \neq \mathsf{ct}_B^*$, we must have $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) = \bot$. This is because, as can be seen from the description of $\overline{\mathsf{Dec}}$ above, we have $\mathsf{ct}_{in}^* \leftarrow \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A^*)$ and $(r_A, r_B, \mathsf{m}_b) \leftarrow \mathsf{Dec}_{qdcca}(\mathsf{sk}_{in}, \mathsf{ct}_{in}^*)$ based on PKE_{1-cca}'s correctness. However, the *re-encryption* check $\mathsf{Enc}_{cpa}(\mathsf{pk}_B, \mathsf{ct}_{in}^*; r_B) \neq \mathsf{ct}_B$ fails in the $\overline{\mathsf{Dec}}$ algorithm which leads to $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) = \bot$.

If $z = 1$, we have $\mathsf{ct}_{in}^* \leftarrow \mathsf{Enc}_{qdcca}(\mathsf{pk}_{in}, 0^\ell)$ and $\mathsf{ct}_A^* = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}_{in}^*; r_A)$. Consider any $\mathsf{ct} = (\mathsf{ct}_A, \mathsf{ct}_B)$ with $\mathsf{ct}_A = \mathsf{ct}_A^*$ and $\mathsf{ct}_B \neq \mathsf{ct}_B^*$. When decrypting ct according to $\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)$, we have $\mathsf{ct}_{in}^* \leftarrow \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A^*)$ and $0^\ell \leftarrow \mathsf{Dec}_{qdcca}(\mathsf{sk}_{in}, \mathsf{ct}_{in}^*)$ from PKE_{1-cca}'s correctness. Observe that in order to have $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) \neq \bot$, the re-encryption check $\mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}_{in}^*; 0^{|r_A|}) = \mathsf{ct}_A^*$ must be satisfied. However, we know from the γ-spreadness of PKE_{1-cca} that $\mathsf{ct}_A^* = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}_{in}^*; r_A)$ comes from an entropy source of γ-bits, for a uniform $r_A \leftarrow \{0,1\}^\lambda$. Hence, the check $\mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}_{in}^*; 0^{|r_A|}) = \mathsf{ct}_A^*$ is satisfied with probability at most $2^{-\lambda}$. We hence obtain $|\Pr[W^{(1)}] - \Pr[W^{(2)}]| \leq 2^{-\gamma}$. □

Lemma 7.4. *There exists a QPT adversary $\mathcal{B}_{ind}$ such that* $|\Pr[W^{(3)}] - 1/2| = \mathbf{Adv}^{\text{qInd}}_{\mathsf{DPKE}, \mathcal{B}_{ind}}$.

Proof. On input pk_{in}, first $\mathcal{B}_{ind}$ generates two pairs $(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{Gen}_{1-cca}(1^\lambda)$ and $(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{Gen}_{cpa}(1^\lambda)$. It then sets $\mathsf{pk} = (\mathsf{pk}_{in}, \mathsf{pk}_A, \mathsf{pk}_B)$ and forwards pk to $\mathcal{A}$. Next, $\mathcal{B}_{ind}$ proceeds to simulate quantum access to the oracle $|\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)\rangle$ for $\mathcal{A}$ using the self-generated pairs $(\mathsf{pk}_A, \mathsf{sk}_A)$, $(\mathsf{pk}_B, \mathsf{sk}_B)$, and its own *quantum* oracle $|\mathsf{Dec}_{qdcca}(\mathsf{sk}_{in}, \cdot)\rangle$ in a straightforward manner. Upon receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, the adversary $\mathcal{B}_{ind}$ samples $b \leftarrow \{0,1\}$ and $r_A, r_B \leftarrow \{0,1\}^\lambda$. It then sends $((r_A, r_B, \mathsf{m}_b), 0^\ell)$, where ℓ is the bit-length of (r_A, r_B, m_b)), to its own challenger. After receiving ct^*_{in}, $\mathcal{B}_{ind}$ computes $\mathsf{ct}^*_A = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}^*_{in}; r_A)$, $\mathsf{ct}^*_B = \mathsf{Enc}_{cpa}(\mathsf{pk}_B, \mathsf{ct}^*_{in}; r_B)$, and returns the ciphertext $\mathsf{ct}^* = (\mathsf{ct}^*_A, \mathsf{ct}^*_B)$ to $\mathcal{A}$. Next, $\mathcal{B}_{ind}$ proceeds to respond to the rest of $\mathcal{A}$'s queries as in the pre-challenge phase. Since $\mathcal{B}_{ind}$ now has quantum access to a *restricted* decryption oracle $|O^F_{\mathsf{ct}^*_{in}}(\mathsf{sk}_{in}, \cdot)\rangle$ in the post-challenge phase (Definition 7.1), it perfectly simulates the *modified* post-challenge decryption oracle $|\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)\rangle$ of Game 3 towards $\mathcal{A}$. Specifically, for any $|\mathsf{ct}\rangle = |(\mathsf{ct}_A, \mathsf{ct}_B)\rangle$ in the computational basis, when $\mathcal{B}_{ind}$ computes $\mathsf{ct}_{in} = \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A)$ and forwards ct_{in} to its quantum oracle $|O^F_{\mathsf{ct}^*_{in}}(\mathsf{sk}_{in}, \cdot)\rangle$, it gets $\perp$ if either $\mathsf{ct}_A = \mathsf{ct}^*_A$ or $F(\mathsf{pk}_{in}, \mathsf{ct}^*_{in}, \mathsf{ct}_{in}) = 1$. Finally, when $\mathcal{A}$ outputs a bit z', $\mathcal{B}_{ind}$ outputs the same z' as well. Note that $\mathcal{B}_{ind}$ perfectly simulates Game 3 towards $\mathcal{A}$. Hence, $\mathcal{B}_{ind}$ wins its game if and only if $\mathcal{A}$ wins Game 3, which complete the proof. □

Now all that remains towards proving the Nest-qCCA security of $\overline{\mathsf{PKE}}$ is to bound $|\Pr[W^{(2)}] - \Pr[W^{(3)}]|$. In the context of Lemma 2.1, let G (respectively, H) be the post-challenge decryption oracle in Game 2 (respectively, Game 3). Note that the outputs of G and H differ with respect to ciphertexts $\mathsf{ct} = (\mathsf{ct}_A, \mathsf{ct}_B)$, where $\mathsf{ct}_A \neq \mathsf{ct}^*_A$, such that $G(\mathsf{ct}) \neq \perp$ and $H(\mathsf{ct}) = \perp$, i.e., these ciphertexts $\mathsf{ct} = (\mathsf{ct}_A, \mathsf{ct}_B)$ must satisfy $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) \neq \perp$, $\mathsf{ct}_A \neq \mathsf{ct}^*_A$, and $F(\mathsf{pk}_{in}, \mathsf{ct}^*_{in}, \mathsf{ct}_{in}) = 1$ where $\mathsf{ct}_{in} = \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A)$. Ignoring the requirement of $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) \neq \perp$, we call any ct that satisfies the remaining two conditions (i.e., $\mathsf{ct}_A \neq \mathsf{ct}^*_A$ and $F(\mathsf{pk}_{in}, \mathsf{ct}^*_{in}, \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A)) = 1$) a *bad-query* (borrowing from [HLW12]). The reason we ignore the $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) \neq \perp$ requirement will be made clear towards the end of our qCCA security proof.

We now define $M^{(j)}$, for $j \in [6]$, to be the event when the measurement of a random i-th quantum decryption query made by $\mathcal{A}$ in the post-challenge phase of Game j (where $i \leftarrow [q_{post}]$) results in a bad-query. Based on our usage of Lemma 2.1 in the qCCA security proofs of previous sections (particularly when modifying a decryption oracle in the post-challenge phase, e.g., in Lemma 3.6 and Lemma 6.3), it is not hard to see that we have $|\Pr[W^{(2)}] - \Pr[W^{(3)}]| \leq 2q_{post}\sqrt{\Pr[M^{(3)}]}$.[1]

Towards bounding $\Pr[M^{(3)}]$, we bound the probability *conditioned* on the hidden bit z of Game 3 being 0 or 1. Here, we will be adopting a deferred analysis approach as in [HLW12, Section 4.1]. Namely, we show the following:

Lemma 7.5. *There exists a QPT adversary $\mathcal{B}_{cpa}$ such that*

$$|\Pr[M^{(3)}|z = 1] - \Pr[M^{(4)}|z = 1]| = 2 \cdot \mathbf{Adv}^{\mathsf{CPA}}_{\mathsf{PKE}_{cpa}, \mathcal{B}_{cpa}}.$$

Proof. On input pk_B, first $\mathcal{B}_{cpa}$ generates key pairs $(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{Gen}_{1-cca}(1^\lambda)$ and $(\mathsf{pk}_{in}, \mathsf{sk}_{in}) \leftarrow \mathsf{Gen}_{qdcca}(1^\lambda)$. It then sets $\mathsf{pk} = (\mathsf{pk}_{in}, \mathsf{pk}_A, \mathsf{pk}_B)$ and sends pk to $\mathcal{A}$. Next, $\mathcal{B}_{cpa}$ proceeds to simulate quantum access to the oracle $|\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)\rangle$ for $\mathcal{A}$ using sk_A and sk_{in} in the usual manner. Upon receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, the adversary $\mathcal{B}_{cpa}$ chooses $r_A \leftarrow \{0,1\}^\lambda$, and computes $\mathsf{ct}^*_{in} \leftarrow \mathsf{Enc}_{qdcca}(\mathsf{pk}_{in}, 0^\ell)$ and $\mathsf{ct}^*_A = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}^*_{in}; r_A)$. Afterwards, $\mathcal{B}_{cpa}$ sends $(\mathsf{ct}^*_{in}, 1^k)$ to its own challenger. After receiving ct^*_B, the adversary $\mathcal{B}_{cpa}$ returns $\mathsf{ct}^* = (\mathsf{ct}^*_A, \mathsf{ct}^*_B)$ to $\mathcal{A}$. Next, $\mathcal{B}_{cpa}$ samples $i \leftarrow [q_{post}]$ and proceeds to respond to the rest of $\mathcal{A}$'s queries according to the post-challenge decryption oracle in Game 3 (and 4) *until* the i-th query. Specifically, before the i-th query, with respect to $\mathcal{A}$'s queried ciphertext $|\mathsf{ct}\rangle = |(\mathsf{ct}_A, \mathsf{ct}_B)\rangle$ in the computational basis, $\mathcal{B}_{cpa}$ returns $\perp$ if either $\mathsf{ct}_A = \mathsf{ct}^*_A$ or $F(\mathsf{pk}_{in}, \mathsf{ct}^*_{in}, \mathsf{ct}_{in}) = 1$ where $\mathsf{ct}_{in} = \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A)$; otherwise, $\mathcal{B}_{cpa}$ decrypts using $\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)$. When $\mathcal{A}$ makes the i-th query, $\mathcal{B}_{cpa}$ measures it with the resulting state being $\overline{\mathsf{ct}} = (\overline{\mathsf{ct}}_A, \overline{\mathsf{ct}}_B)$ and checks if $\overline{\mathsf{ct}}$ is a *bad-query* as defined above. If $\overline{\mathsf{ct}}$ is a bad-query, $\mathcal{B}_{cpa}$ outputs 1; otherwise, it outputs 0.

Note that $\mathcal{B}_{cpa}$ simulates Game 3 towards $\mathcal{A}$ (conditioned on $z = 1$) if its challenger encrypted the "left" message ct^*_{in}. Similarly, $\mathcal{B}_{cpa}$ simulates Game 4 (conditioned on $z = 1$) if its challenger encrypted the "right" message 1^k. Hence, if we let b to be the hidden bit chosen by $\mathcal{B}_{cpa}$'s challenger then we have

$$\mathbf{Adv}^{\mathsf{CPA}}_{\mathsf{PKE}_{cpa}, \mathcal{B}_{cpa}}(\lambda) = \frac{1}{2} \cdot \Big| \Pr[1 \leftarrow \mathcal{B}_{cpa}|b = 0] - \Pr[1 \leftarrow \mathcal{B}_{cpa}|b = 1] \Big| = \frac{1}{2} \cdot \Big| \Pr[M^{(3)}|z = 1] - \Pr[M^{(4)}|z = 1] \Big|.$$

□

[1] Technically, in the context of applying Lemma 2.1, the probability P_{guess} corresponds to the measured ciphertext *also* satisfying $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) \neq \perp$, in addition to being a bad-query. But we have P_{guess} to be trivially upper-bounded by $\Pr[M^{(3)}]$.

Lemma 7.6. *There exists a QPT adversary $\mathcal{B}_{1-cca}$ such that*

$$|\Pr[M^{(4)}|z=1] - \Pr[M^{(5)}|z=1]| = 2 \cdot \mathbf{Adv}^{\text{1-CCA}}_{\mathsf{PKE},\mathcal{B}_{1-cca}}.$$

Proof. On input pk_A, first $\mathcal{B}_{1-cca}$ generates key pairs $(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{Gen}_{cpa}(1^\lambda)$ and $(\mathsf{pk}_{in}, \mathsf{sk}_{in}) \leftarrow \mathsf{Gen}_{qdcca}(1^\lambda)$. It sets $\mathsf{pk} = (\mathsf{pk}_{in}, \mathsf{pk}_A, \mathsf{pk}_B)$ and forwards pk to $\mathcal{A}$. Afterwards, $\mathcal{B}_{1-cca}$ responds to $\mathcal{A}$'s quantum decryption queries as follows: for a ciphertext $|\mathsf{ct}\rangle = |(\mathsf{ct}_A, \mathsf{ct}_B)\rangle$, first $\mathcal{B}_{1-cca}$ computes $\mathsf{ct}_{in} \leftarrow \mathsf{Dec}_{cpa}(\mathsf{sk}_B, \mathsf{ct}_B)$ and $(r_A, r_B, \mathsf{m}) \leftarrow \mathsf{Dec}_{qdcca}(\mathsf{sk}_{in}, \mathsf{ct}_{in})$; $\mathcal{B}'_{cpa}$. It returns the message m if $\mathsf{ct}_A = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}_{in}; r_A)$ and $\mathsf{ct}_B = \mathsf{Enc}_{cpa}(\mathsf{pk}_B, \mathsf{ct}_{in}; r_B)$, otherwise it returns $\perp$. As observed in [HLW12], replacing $\mathsf{ct}_{in} \leftarrow \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A)$ in $\overline{\mathsf{Dec}}$ with $\mathsf{ct}_{in} \leftarrow \mathsf{Dec}_{cpa}(\mathsf{sk}_B, \mathsf{ct}_B)$ does not make any difference. Namely, if $\mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A) = \mathsf{Dec}_{cpa}(\mathsf{sk}_B, \mathsf{ct}_B)$, then both decryption algorithms proceed identically. If $\mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A) \neq \mathsf{Dec}_{cpa}(\mathsf{sk}_B, \mathsf{ct}_B)$, then both algorithms return $\perp$ because the checks $\mathsf{ct}_A = \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, \mathsf{ct}_{in}; r_A)$ and $\mathsf{ct}_B = \mathsf{Enc}_{cpa}(\mathsf{pk}_B, \mathsf{ct}_{in}; r_B)$ for a common ct_{in} will not be simultaneously satisfied. Hence, even if $\mathcal{B}_{1-cca}$ first decrypts ct_B instead of ct_A when responding to $\mathcal{A}$'s decryption query $|\mathsf{ct}\rangle = |(\mathsf{ct}_A, \mathsf{ct}_B)\rangle$, it still simulates the quantum oracle $|\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)\rangle$ for $\mathcal{A}$.

Upon receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, $\mathcal{B}_{1-cca}$ computes $\mathsf{ct}^*_{in} \leftarrow \mathsf{Enc}_{qdcca}(\mathsf{pk}_{in}, 0^\ell)$ (we are conditioning on $z = 1$) and $\mathsf{ct}^*_B \leftarrow \mathsf{Enc}_{cpa}(\mathsf{pk}_B, 1^k)$. Next, $\mathcal{B}_{1-cca}$ sends $(\mathsf{ct}^*_{in}, 1^k)$ to its challenger. After receiving the challenge ct^*_A, the adversary $\mathcal{B}_{1-cca}$ returns the ciphertext $\mathsf{ct}^* = (\mathsf{ct}^*_A, \mathsf{ct}^*_B)$ to $\mathcal{A}$. Next, $\mathcal{B}_{1-cca}$ samples $i \leftarrow [q_{post}]$ and responds to the rest of $\mathcal{A}$'s queries as follows: for any query $|\mathsf{ct}\rangle = |(\mathsf{ct}_A, \mathsf{ct}_B)\rangle$, the adversary $\mathcal{B}_{1-cca}$ returns $\perp$ if either $\mathsf{ct}_A = \mathsf{ct}^*_A$ or $F(\mathsf{pk}_{in}, \mathsf{ct}^*_{in}, \mathsf{ct}_{in})$ = 1 where $\mathsf{ct}_{in} = \mathsf{Dec}_{cpa}(\mathsf{sk}_B, \mathsf{ct}_B)$; otherwise, it decrypts ct as in the pre-challenge phase above. If $\mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \mathsf{ct}_A) \neq \mathsf{Dec}_{cpa}(\mathsf{sk}_B, \mathsf{ct}_B)$, then both post-challenge decryption algorithms return $\perp$ since $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) = \perp$. Hence, $\mathcal{B}_{1-cca}$ simulates the post-challenge decryption oracle in Game 4 (and 5) for $\mathcal{A}$ *until* the i-th query. When $\mathcal{A}$ makes the i-th query, $\mathcal{B}_{1-cca}$ measures it with the resulting state being $\overline{\mathsf{ct}} = (\overline{\mathsf{ct}}_A, \overline{\mathsf{ct}}_B)$ and checks if $\overline{\mathsf{ct}}$ is a *bad-query* as follows: $\mathcal{B}_{1-cca}$ first checks if $\overline{\mathsf{ct}}_A \neq \mathsf{ct}^*_A$; if so, it forwards $\overline{\mathsf{ct}}_A$ to its *classical* one-time decryption oracle $\mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \cdot)$ to obtain $\overline{\mathsf{ct}}_{in}$ and checks if $F(\mathsf{pk}_{in}, \mathsf{ct}^*_{in}, \overline{\mathsf{ct}}_{in}) = 1$. If the checks pass (i.e., $\overline{\mathsf{ct}}$ is a bad-query) then $\mathcal{B}_{1-cca}$ outputs 1 and otherwise outputs 0.

Observe that $\mathcal{B}_{1-cca}$ simulates Game 4 (respectively, Game 5) towards $\mathcal{A}$ (conditioned on $z = 1$) if its challenger encrypted the "left" (respectively, "right") message, and hence

$$\mathbf{Adv}^{\text{1-CCA}}_{\mathsf{PKE}_{1-cca},\mathcal{B}_{1-cca}} = \frac{1}{2} \cdot \Big| \Pr[M^{(4)}|z=1] - \Pr[M^{(5)}|z=1] \Big|. \qquad \square$$

Before bounding $|\Pr[M^{(5)}|z=1] - \Pr[M^{(6)}|z=1]|$, we show the following for Game 6.

Lemma 7.7. *There exists a QPT adversary $\mathcal{B}_{pred}$ such that*

$$\Pr[M^{(6)}|z=1] \leq \mathbf{Adv}^{\text{qPredict}}_{\mathsf{DPKE},\mathcal{B}_{pred}}.$$

Proof. On input pk_{in}, first $\mathcal{B}_{pred}$ generates $(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{Gen}_{1-cca}(1^\lambda)$ and $(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{Gen}_{cpa}(1^\lambda)$. It then sets $\mathsf{pk} = (\mathsf{pk}_{in}, \mathsf{pk}_A, \underline{\mathsf{pk}_B})$ and forwards pk to $\mathcal{A}$. In the pre-challenge phase, $\mathcal{B}_{pred}$ proceeds to simulate quantum access to the oracle $|\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)\rangle$ towards $\mathcal{A}$ using $(\mathsf{sk}_A, \mathsf{sk}_B)$ and its own *quantum* oracle $|\mathsf{Dec}_{qdcca}(\mathsf{sk}_{in}, \cdot)\rangle$ in a straightforward manner. Upon receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}$, the adversary $\mathcal{B}_{pred}$ computes $\mathsf{ct}^*_A \leftarrow \mathsf{Enc}_{1-cca}(\mathsf{pk}_A, 1^k)$, $\mathsf{ct}^*_B \leftarrow \mathsf{Enc}_{cpa}(\mathsf{pk}_B, 1^k)$, and returns $\mathsf{ct}^* = (\mathsf{ct}^*_A, \mathsf{ct}^*_B)$. Next, $\mathcal{B}_{pred}$ samples $i \leftarrow [q_{post}]$ and responds to $\mathcal{A}$'s queries as in the pre-challenge phase (while rejecting ciphertexts $(\mathsf{ct}^*_A, \mathsf{ct}_B)$) *until* the i-th query. When $\mathcal{A}$ makes the i-th query, $\mathcal{B}_{pred}$ measures it with the resulting state being $\overline{\mathsf{ct}} = (\overline{\mathsf{ct}}_A, \overline{\mathsf{ct}}_B)$. If $\overline{\mathsf{ct}}_A \neq \mathsf{ct}^*_A$, then $\mathcal{B}_{pred}$ computes $\overline{\mathsf{ct}}_{in} = \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \overline{\mathsf{ct}}_A)$ and returns $(0^\ell, \overline{\mathsf{ct}}_{in})$; otherwise, $\mathcal{B}_{pred}$ aborts. Note that $\mathcal{B}_{pred}$ simulates Game 6 until the i-th query for $\mathcal{A}$. Hence, conditioned on $z = 1$, if the event $M^{(6)}$ occurs, it means the measured $\overline{\mathsf{ct}} = (\overline{\mathsf{ct}}_A, \overline{\mathsf{ct}}_B)$ is a bad-query that satisfies $F(\mathsf{pk}_{in}, \mathsf{ct}^*_{in}, \overline{\mathsf{ct}}_{in}) = 1$ where $\mathsf{ct}^*_{in} \leftarrow \mathsf{Enc}_{qdcca}(\mathsf{pk}_{in}, 0^\ell)$. Since $(0^\ell, \overline{\mathsf{ct}}_{in})$ would result in breaking the quantum unpredictability of F, it follows that $\Pr[M^{(6)}|z=1] \leq \mathbf{Adv}^{\text{qPredict}}_{\mathsf{DPKE},\mathcal{B}_{pred}}$. $\square$

Lemma 7.8. *There exists a QPT adversary $\mathcal{B}'_{pred}$ such that*

$$\Big| \Pr[M^{(5)}|z=1] - \Pr[M^{(6)}|z=1] \Big| \leq 2 q_{post} \sqrt{\mathbf{Adv}^{\text{qPredict}}_{\mathsf{DPKE},\mathcal{B}'_{pred}}}.$$

Proof. The proof is quite similar to that of Lemma 7.7, but additionally involves a *nested* application of Lemma 2.1. First, it helps to interpret the probability $\Pr[M^{(j)}|z = 1]$, for $j \in \{5, 6\}$, corresponding to $\mathcal{A}$ winning a *variant* of Game j (conditioned on $z = 1$) which we call $\overline{\text{Game}}\ j$. The *only* difference between Game j and $\overline{\text{Game}}\ j$ is their respective winning conditions. Recall that $\mathcal{A}$ is said to win Game j if it outputs the hidden bit z, and $\mathcal{A}$ wins $\overline{\text{Game}}\ j$ if the measurement of a random i-th quantum decryption query made by $\mathcal{A}$ in the post-challenge phase (where $i \leftarrow [q_{post}]$) results in a *bad-query*. Based on Lemma 2.1, one can construct an oracle algorithm A (which has access either to the post-challenge decryption oracle of Game 5 or Game 6) such that it simulates either $\overline{\text{Game}}$ 5 or $\overline{\text{Game}}$ 6 (conditioned on $z = 1$) for $\mathcal{A}$. Here, another algorithm A' first simulates the pre-challenge phase of Games 5 and 6 towards $\mathcal{A}$, and then forwards the appropriate input to A. Such an algorithm A would output 1 if and only if $\mathcal{A}$ wins the corresponding $\overline{\text{Game}}\ j \in \{5, 6\}$. Note that A can check if a measured ciphertext is a bad-query since A' would forward the relevant secret keys to A.

Let $\overline{M}^{(j)}$ for $j \in \{5, 6\}$ be the event when the measurement of a random $\bar{i}$-th quantum decryption query made by $\mathcal{A}$ in the post-challenge phase of $\overline{\text{Game}}\ j$ (where $\bar{i} \leftarrow [q_{post}]$) results in a bad-query. From a "nested" application of Lemma 2.1 outlined above we get

$$\left|\Pr[M^{(5)}|z = 1] - \Pr[M^{(6)}|z = 1]\right| \leq 2q_{post}\sqrt{\Pr[\overline{M}^{(6)}|z = 1]}.$$

To bound $\Pr[\overline{M}^{(6)}|z = 1]$, we can construct a QPT adversary $\mathcal{B}'_{pred}$ (similar to $\mathcal{B}_{pred}$ in the proof of Lemma 7.7) which breaks the quantum unpredictability of F with respect to DPKE. Note that $\mathcal{B}_{pred}$ above perfectly simulated Game 6, until the i-th post-challenge decryption query for a random $i \leftarrow [q_{post}]$ towards $\mathcal{A}$. In this setting, $\mathcal{B}'_{pred}$ would perfectly simulate $\overline{\text{Game}}$ 6, until the $\bar{i}$-th query for a random $\bar{i} \leftarrow [q_{post}]$ towards $\mathcal{A}$. The only difference between $\mathcal{B}_{pred}$ and $\mathcal{B}'_{pred}$ would be that the latter algorithm samples *two* query numbers $i, \bar{i} \leftarrow [q_{post}]$ in the post-challenge phase of $\overline{\text{Game}}$ 6; the first query number i is what the challenger in $\overline{\text{Game}}$ 6 would have sampled for checking the winning condition with respect to $\mathcal{A}$, and the second query number $\bar{i}$ is what $\mathcal{B}'_{pred}$ is aiming for with respect to measuring $\mathcal{A}$'s post-challenge query and breaking the quantum unpredictability of F via a nested application of Lemma 2.1. If $\bar{i} > i$, then $\mathcal{B}'_{pred}$ would return $\perp$; otherwise, it would measure the $\bar{i}$-th post-challenge query to a state $\overline{\mathsf{ct}} = (\overline{\mathsf{ct}}_A, \overline{\mathsf{ct}}_B)$ and return $(0^\ell, \mathsf{Dec}_{1-cca}(\mathsf{sk}_A, \overline{\mathsf{ct}}_A))$ to its challenger if $\overline{\mathsf{ct}}_A \neq \mathsf{ct}^*_A$ (similar to $\mathcal{B}_{pred}$ above).[1]

Observe that conditioned on $z = 1$, if the event $\overline{M}^{(6)}$ occurs, then the result of $\mathcal{B}'_{pred}$'s measurement in the post-challenge phase is a bad-query, which means breaking the quantum unpredictability of F as described above. It follows that $\Pr[\overline{M}^{(6)}|z = 1] \leq \mathbf{Adv}^{\mathsf{qPredict}}_{\mathsf{DPKE},\mathcal{B}'_{pred}}$, as desired. □

Based on the lemmas above, it follows that $\Pr[M^{(3)}|z = 1]$ is bounded by a negligible quantity. To show that $\Pr[M^{(3)}]$ is also negligible, we bound $\Pr[M^{(3)}|z = 0]$ by relying on the qDCCA security of DPKE as follows.

Lemma 7.9. *There exists a QPT adversary $\mathcal{B}'_{ind}$ such that*

$$|\Pr[M^{(3)}|z = 0] - \Pr[M^{(3)}|z = 1]| = 2 \cdot \mathbf{Adv}^{\mathsf{qInd}}_{\mathsf{DPKE},\mathcal{B}'_{ind}}.$$

Proof. The description of $\mathcal{B}'_{ind}$ is quite similar to that of $\mathcal{B}_{ind}$ in Lemma 7.4. Both algorithms proceed *identically* in the pre-challenge and challenge phases. The only difference is in the post-challenge phase where, after forwarding the challenge ciphertext $\mathsf{ct}^* = (\mathsf{ct}^*_A, \mathsf{ct}^*_B)$ to $\mathcal{A}$, the adversary $\mathcal{B}'_{ind}$ samples $i \leftarrow [q_{post}]$ and responds to $\mathcal{A}$'s quantum decryption queries *until* the i-th query using the self-generated key-pairs $(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{Gen}_{1-cca}(1^\lambda)$, $(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{Gen}_{cpa}(1^\lambda)$, and its own post-challenge oracle $|O^F_{\mathsf{ct}^*_{in}}(\mathsf{sk}_{in}, \cdot)\rangle$ (Definition 7.1). Observe that $\mathcal{B}'_{ind}$ perfectly simulates the post-challenge decryption oracle of Game 3 towards $\mathcal{A}$ until the i-th query. When $\mathcal{A}$ makes the i-th query, $\mathcal{B}'_{ind}$ measures it with the resulting state being $\overline{\mathsf{ct}} = (\overline{\mathsf{ct}}_A, \overline{\mathsf{ct}}_B)$ and checks if $\overline{\mathsf{ct}}$ is a *bad-query*. If $\overline{\mathsf{ct}}$ is a bad-query, $\mathcal{B}'_{ind}$ outputs 1; otherwise, it outputs 0.

[1] In the context of Lemma 2.1, note that the case $\bar{i} > i$ translates to the setting when the oracle algorithm A (simulating $\overline{\text{Game}}$ 6 towards $\mathcal{A}$) makes less than $\bar{i}$ queries to its quantum oracle, and hence is accounted for by the generalized OW2H lemma.

Note that $\mathcal{B}'_{ind}$ simulates Game 3 towards $\mathcal{A}$ until its i-th post-challenge decryption query where the hidden bit z in Game 3 can be interpreted as the *same* hidden bit sampled by $\mathcal{B}'_{ind}$'s challenger. Therefore, we have

$$\mathbf{Adv}^{\text{qInd}}_{\mathsf{DPKE},\mathcal{B}'_{ind}} = \frac{1}{2} \cdot \left| \Pr[1 \leftarrow \mathcal{B}'_{ind} | z = 0] - \Pr[1 \leftarrow \mathcal{B}'_{ind} | z = 1] \right| = \frac{1}{2} \cdot \left| \Pr[M^{(3)} | z = 0] - \Pr[M^{(3)} | z = 1] \right|.$$

Remark 7.10. Note that if we included the requirement of $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) \neq \bot$ in our definition of bad-queries, there would have been an issue in the reduction above. Specifically, for the post-measurement ciphertext $\overline{\mathsf{ct}}$, if we have $\overline{\mathsf{ct}}_A \neq \mathsf{ct}^*_A$ and $F(\mathsf{pk}_{in}, \mathsf{ct}^*_{in}, \overline{\mathsf{ct}}_{in}) = 1$, then $\mathcal{B}'_{ind}$ could not check $\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct}) \neq \bot$ because its post-challenge oracle $|O^F_{\mathsf{ct}^*_{in}}(\mathsf{sk}_{in}, \cdot)\rangle$ would have forbidden a query on $\overline{\mathsf{ct}}_{in}$.

Since $\Pr[M^{(3)} | z = 0]$ and $\Pr[M^{(3)} | z = 1]$ are negligible, it follows that $\Pr[M^{(3)}]$ is also negligible. By relying on Lemma 7.3 and Lemma 7.4, and using the fact that $|\Pr[W^{(2)}] - \Pr[W^{(3)}]| \leq 2q_{post}\sqrt{\Pr[M^{(3)}]}$, it follows that

$$|\Pr[W^{(1)}] - 1/2| = \mathbf{Adv}^{\text{Nest-qInd}}_{(\mathsf{PKE}_{1-cca}, \mathsf{PKE}_{cpa}, \mathsf{DPKE}), \mathcal{A}} \leq \text{negl},$$

which establishes the quantum CCA security of $\overline{\mathsf{PKE}}$ and complete our proof of Theorem 7.2. □

7.2 Multi-Bit Quantum DCCA Security from Single-Bit Quantum CCA Security

After showing that multi-bit qDCCA-secure detectable PKE implies multi-bit qCCA-secure PKE, we now describe how to realize qDCCA-secure PKE from single-bit qCCA-secure PKE, proving the completeness of bit encryption for qCCA security. Let $\mathsf{PKE}_{1-bit} = (\mathsf{Gen}_{1-bit}, \mathsf{Enc}_{1-bit}, \mathsf{Dec}_{1-bit})$ be a single-bit qCCA-secure PKE with perfect correctness,[1] and also let F^* be a detecting function that outputs 0 on all inputs except those of the form $(\cdot, \mathsf{ct}, \mathsf{ct})$. In the first step, we show that $\mathsf{DPKE}_{1-bit} = (\mathsf{Gen}_{1-bit}, \mathsf{Enc}_{1-bit}, \mathsf{Dec}_{1-bit}, F^*)$ is a single-bit detectable PKE scheme with qDCCA security.

Lemma 7.11. *If PKE_{1-bit} is a (single-bit) qCCA-secure PKE scheme then DPKE_{1-bit} satisfies qDCCA security.*

Proof. Quantum indistinguishability of encryptions (Definition 7.1) can be shown by a straightforward reduction. To show quantum unpredictability of F^*, given a QPT adversary $\mathcal{A}$ which breaks the quantum unpredictability of F^* with advantage ε, we describe an adversary $\mathcal{A}'$ which breaks the qCCA security of PKE_{1-bit} with advantage $\varepsilon/4$. On input pk, first $\mathcal{A}'$ forwards pk to $\mathcal{A}$. It simulates quantum access to the oracle $|\mathsf{Dec}_{1-bit}(\mathsf{sk}, \cdot)\rangle$ towards $\mathcal{A}$ as $\mathcal{A}'$ has access to the *same* oracle in the pre-challenge phase. When $\mathcal{A}$ outputs a pair $(\mathsf{m}, \mathsf{ct})$, the adversary $\mathcal{A}'$ first queries ct to its oracle $|\mathsf{Dec}_{1-bit}(\mathsf{sk}, \cdot)\rangle$ in the pre-challenge phase, and records the response as $\mathsf{m}' = \mathsf{Dec}_{1-bit}(\mathsf{sk}, \mathsf{ct})$. Then $\mathcal{A}'$ forwards $(\mathsf{m}, 1 - \mathsf{m})$ to its challenger. Upon receiving ct^*, the adversary $\mathcal{A}'$ first checks if $\mathsf{m}' \in \{1 - \mathsf{m}, \bot\}$. If so, $\mathcal{A}'$ outputs a random bit b'. Otherwise, if $\mathsf{m}' = \mathsf{m}$, then $\mathcal{A}'$ further checks if $\mathsf{ct} = \mathsf{ct}^*$. If so, $\mathcal{A}'$ guesses that the "left" message m was encrypted ($b' = 0$); otherwise, $\mathcal{A}'$ outputs a random bit b'.

Since $\mathcal{A}$ breaks the quantum unpredictability of F^* with advantage ε, we have that $F^*(\mathsf{pk}, \mathsf{ct}^*, \mathsf{ct}) = 1$ with probability ε, where $\mathsf{ct}^* \leftarrow \mathsf{Enc}_{1-bit}(\mathsf{pk}, \mathsf{m})$. Hence in the qCCA game, if the challenger encrypted the "left" message m, then $\mathcal{A}'$ outputs $b' = 0$ with probability $\varepsilon + (1 - \varepsilon)/2$. If the qCCA challenger encrypted the "right" message $1 - \mathsf{m}$ instead, then note that $\mathcal{A}'$'s check $\mathsf{ct} = \mathsf{ct}^*$ above is never satisfied. Hence when $b = 1$, the adversary $\mathcal{A}'$ outputs a random bit b'. Therefore, we have

$$\begin{aligned}\mathbf{Adv}^{\text{qCCA}}_{\mathsf{PKE}_{1-bit}, \mathcal{A}'} &= \frac{1}{2} \cdot \left| \Pr[b' = 0 | b = 0] - \Pr[b' = 0 | b = 1] \right| \\ &= \frac{1}{2} \cdot \left| \varepsilon + \frac{1 - \varepsilon}{2} - \frac{1}{2} \right| = \frac{\varepsilon}{4}.\end{aligned}$$

□

Now starting with *any* single-bit qDCCA-secure detectable PKE scheme $\mathsf{DPKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, F)$ where F can be arbitrary and need not be equal to F^* above, consider the following construction of a multi-bit detectable PKE scheme $\overline{\mathsf{DPKE}} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}, \overline{F})$:[2]

[1] As noted previously, we make this assumption for the ease of exposition. One can extend our analysis to the case where PKE_{1-bit} satisfies almost-all-keys correctness.

[2] This is essentially the same construction as in [HLW12, Appendix B] (for classical CCA security).

$\overline{\mathsf{Gen}}(1^\lambda)$: Sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and output $(\mathsf{pk}, \mathsf{sk})$.

$\overline{\mathsf{Enc}}(\mathsf{pk}, \mathsf{m})$: Let $n = |\mathsf{m}|$. For all $i \in [n]$, compute $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_i)$ where $\mathsf{m}_i \in \{0, 1\}$, and output $\mathsf{ct} = \mathsf{ct}_1 \parallel \mathsf{ct}_2 \parallel \ldots \parallel \mathsf{ct}_n$.

$\overline{\mathsf{Dec}}(\mathsf{sk}, \mathsf{ct})$: Parse $\mathsf{ct} \rightarrow \mathsf{ct}_1 \parallel \ldots \parallel \mathsf{ct}_n$. For each $i \in [n]$, compute $\mathsf{m}_i = \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_i)$. If $\mathsf{m}_i = \perp$ for some $i \in [n]$, return $\perp$. Otherwise, output $\mathsf{m} = \mathsf{m}_1 \parallel \mathsf{m}_2 \parallel \ldots \parallel \mathsf{m}_n$.

$\overline{F}(\mathsf{pk}, \mathsf{ct}', \mathsf{ct})$: Parse $\mathsf{ct}' = \mathsf{ct}'_1 \parallel \ldots \parallel \mathsf{ct}'_n$ and $\mathsf{ct} = \mathsf{ct}_1 \parallel \ldots \parallel \mathsf{ct}_n$. If there is a pair $(i, j) \in [n]^2$ such that $F(\mathsf{pk}, \mathsf{ct}'_i, \mathsf{ct}_j) = 1$, return 1. Otherwise, return 0.

Theorem 7.12. *If* DPKE *is a (single-bit) qDCCA-secure detectable PKE then* $\overline{\mathsf{DPKE}}$ *is a (multi-bit) qDCCA-secure detectable PKE.*

Proof. The proof follows quite closely the proof of [HLW12, Appendix B]. We first show the *quantum* unpredictability of $\overline{F}$ while relying on the quantum unpredictability of F. Given a QPT adversary $\mathcal{A}$ with advantage ε, we construct a QPT adversary $\mathcal{A}'$ as follows. On input pk, first $\mathcal{A}'$ forwards pk to $\mathcal{A}$. It then simulates quantum access to the oracle $|\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)\rangle$ towards $\mathcal{A}$ using its own quantum oracle $|\mathsf{Dec}(\mathsf{sk}, \cdot)\rangle$ as follows. For any queried ciphertext $|\mathsf{ct}\rangle$ in the computational basis, $\mathcal{A}'$ parses it as $\mathsf{ct} = \mathsf{ct}_1 \parallel \ldots \parallel \mathsf{ct}_n$ *in a reversible way*. Then $\mathcal{A}'$ queries each ct_i, for $i \in [n]$, to its oracle $|\mathsf{Dec}(\mathsf{sk}, \cdot)\rangle$ in a sequence and then concatenates the resulting outputs as $\mathsf{m} = \mathsf{m}_1 \parallel \mathsf{m}_2 \parallel \ldots \parallel \mathsf{m}_n$, again in a reversible way (while rejecting ciphertexts ct as appropriate). When $\mathcal{A}$ finally outputs $(\mathsf{m}, \mathsf{ct})$ with $\mathsf{m} = \mathsf{m}_1 \parallel \ldots \parallel \mathsf{m}_m$ and $\mathsf{ct} = \mathsf{ct}_1 \parallel \ldots \parallel \mathsf{ct}_n$, the adversary $\mathcal{A}'$ samples $i \leftarrow [m]$, $j \leftarrow [n]$, and outputs the pair $(\mathsf{m}_i, \mathsf{ct}_j)$ to its challenger. To analyze the advantage of $\mathcal{A}'$, the corresponding analysis of $\overline{F}$'s *classical* unpredictability in [HLW12] extends to the quantum setting in an identical fashion, and it follows that the advantage of $\mathcal{A}'$ is at least ε/mn.

Now we show the quantum indistinguishability of encryptions of $\overline{\mathsf{DPKE}}$ while relying on the same property of DPKE. Given a QPT adversary $\mathcal{B}$ with advantage ε, we construct a QPT adversary $\mathcal{B}'$ as follows. Let n be the length of messages $(\mathsf{m}_0, \mathsf{m}_1)$ output by $\mathcal{B}$ in the challenge phase. Let $\mathsf{m}_0 = \mathsf{m}_{0,1} \parallel \mathsf{m}_{0,2} \ldots \parallel \mathsf{m}_{0,n}$ and $\mathsf{m}_1 = \mathsf{m}_{1,1} \parallel \mathsf{m}_{1,2} \ldots \parallel \mathsf{m}_{1,n}$. We define Games 1 to n as follows, where each Game i is same as the quantum indistinguishability of encryptions game with respect to $\overline{\mathsf{DPKE}}$ (Definition 7.1) except for how the challenge ciphertext ct^* is computed. In Game i, ct^* is computed as

$$\mathsf{ct}^*_j = \begin{cases} \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_{0,j}) & \text{if } 1 \leq j < i, \\ \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_{b,j}) & \text{if } j = i, \\ \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_{1,j}) & \text{if } i < j \leq n, \end{cases}$$

where b is the challenge bit. By a standard hybrid argument, observe that there exist an index i^* where the advantage of $\mathcal{B}$ in Game i^* should be at least ε/n. We now construct a QPT adversary $\mathcal{B}'$ as follows: on input pk, the adversary $\mathcal{B}'$ forwards pk to $\mathcal{B}$. It then simulates quantum access to the oracle $|\overline{\mathsf{Dec}}(\mathsf{sk}, \cdot)\rangle$ towards $\mathcal{B}$ in the pre-challenge phase using its pre-challenge quantum oracle $|\mathsf{Dec}(\mathsf{sk}, \cdot)\rangle$ in the same way as the adversary $\mathcal{A}$. When $\mathcal{B}$ forwards $(\mathsf{m}_0, \mathsf{m}_1)$ in the challenge phase, $\mathcal{B}'$ computes ct^* as is done in Game i^*. Specifically, for $1 \leq i < i^*$, $\mathcal{B}'$ computes $\mathsf{ct}^*_i \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_{0,i})$, and for $i^* < i \leq n$, it computes $\mathsf{ct}^*_i \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_{1,i})$. Next, $\mathcal{B}'$ forwards the pair $(\mathsf{m}_{0,i^*}, \mathsf{m}_{1,i^*})$ to its challenger and gets back the ciphertext $\mathsf{ct}^*_{i^*}$. It then returns $\mathsf{ct}^* = \mathsf{ct}^*_1 \parallel \ldots \parallel \mathsf{ct}^*_n$ to $\mathcal{B}$. Finally, $\mathcal{B}'$ proceeds to respond to $\mathcal{B}$'s remaining quantum decryption queries in the post-challenge phase as follows: for any queried ciphertext $|\mathsf{ct}\rangle$, first $\mathcal{B}'$ parses it as $\mathsf{ct} = \mathsf{ct}_1 \parallel \ldots \parallel \mathsf{ct}_m$ and checks if $\overline{F}(\mathsf{pk}, \mathsf{ct}^*, \mathsf{ct}) = 1$ (using the description of $\overline{F}$ above). If so, $\mathcal{B}'$ returns $\perp$; otherwise, it decrypts ct in the same way as in the pre-challenge phase. When $\mathcal{B}$ terminates with a bit b', the adversary $\mathcal{B}'$ outputs the same bit b'. Observe that $\mathcal{B}'$ perfectly simulates Game i^* towards $\mathcal{B}$ and hence breaks the quantum indistinguishability of encryptions of DPKE with the same advantage that $\mathcal{B}$ has in Game i^*, as desired. □

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Heidelberg, May / June 2010. (Cited on page 2)

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Heidelberg, August 2009. (Cited on page 3)

[ADMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 411–439. Springer, Heidelberg, December 2020. (Cited on page 2, 3, 10, 15)

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 269–295. Springer, Heidelberg, August 2019. (Cited on page 5, 6, 11)

[AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 208–236. Springer, Heidelberg, August 2022. (Cited on page 2)

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. (Cited on page 6)

[BBC$^+$21] Ritam Bhaumik, Xavier Bonnetain, André Chailloux, Gaëtan Leurent, María Naya-Plasencia, André Schrottenloher, and Yannick Seurin. QCB: Efficient quantum-secure authenticated encryption. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 668–698. Springer, Heidelberg, December 2021. (Cited on page 6)

[BCHK07] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007. (Cited on page 2)

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. On the round complexity of secure quantum computation. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 406–435, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 2)

[BDF$^+$11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011. (Cited on page 6)

[BGH$^+$23] Khashayar Barooti, Alex B. Grilo, Loïs Huguenin-Dumittan, Giulio Malavolta, Or Sattath, Quoc-Huy Vu, and Michael Walter. Public-key encryption with quantum keys. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part IV*, volume 14372 of *Lecture Notes in Computer Science*, pages 198–227. Springer, 2023. (Cited on page 4)

[BJSW16] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for QMA. In Irit Dinur, editor, *57th FOCS*, pages 31–40. IEEE Computer Society Press, October 2016. (Cited on page 2)

[BZ13a] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 592–608. Springer, Heidelberg, May 2013. (Cited on page 2)

[BZ13b] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, August 2013. (Cited on page 2, 4, 5, 7, 25)

[CEV22] Céline Chevalier, Ehsan Ebrahimi, and Quoc Huy Vu. On security notions for encryption in a quantum world. In Takanori Isobe and Santanu Sarkar, editors, *INDOCRYPT 2022*, volume 13774 of *LNCS*, pages 592–613. Springer, 2022. (Cited on page 4)

[CHH$^+$07] Ronald Cramer, Goichiro Hanaoka, Dennis Hofheinz, Hideki Imai, Eike Kiltz, Rafael Pass, abhi shelat, and Vinod Vaikuntanathan. Bounded CCA2-secure encryption. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 502–518. Springer, Heidelberg, December 2007. (Cited on page 30)

[CLM$^+$18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018. (Cited on page 2)

[Col23] Andrea Coladangelo. Quantum trapdoor functions from classical one-way functions. Cryptology ePrint Archive, Report 2023/282, 2023. `https://eprint.iacr.org/2023/282`. (Cited on page 4)

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, April / May 2002. (Cited on page 2, 3, 10, 11, 15)

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003. (Cited on page 3, 21)

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *23rd ACM STOC*, pages 542–552. ACM Press, May 1991. (Cited on page 2)

[DFMS22] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 677–706. Springer, Heidelberg, May / June 2022. (Cited on page 6)

[FKS$^+$13] Serge Fehr, Jonathan Katz, Fang Song, Hong-Sheng Zhou, and Vassilis Zikas. Feasibility and completeness of cryptographic tasks in the quantum world. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 281–296. Springer, Heidelberg, March 2013. (Cited on page 2)

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013. (Cited on page 6, 30)

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989. (Cited on page 8, 25)

[HKW20] Susan Hohenberger, Venkata Koppula, and Brent Waters. Chosen ciphertext security from injective trapdoor functions. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 836–866. Springer, Heidelberg, August 2020. (Cited on page 2)

[HLLG19] Shuai Han, Shengli Liu, Lin Lyu, and Dawu Gu. Tight leakage-resilient CCA-security from quasi-adaptive hash proof system. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 417–447. Springer, Heidelberg, August 2019. (Cited on page 2)

[HLW12] Susan Hohenberger, Allison B. Lewko, and Brent Waters. Detecting dangerous queries: A new approach for chosen ciphertext security. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 663–681. Springer, Heidelberg, April 2012. (Cited on page 4, 6, 25, 29, 30, 32, 33, 35, 36)

[KMO10] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 673–692. Springer, Heidelberg, May / June 2010. (Cited on page 2, 3, 4, 25, 28, 29)

[KMP14] Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. Simple chosen-ciphertext security from low-noise LPN. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 1–18. Springer, Heidelberg, March 2014. (Cited on page 2)

[KMT19] Fuyuki Kitagawa, Takahiro Matsuda, and Keisuke Tanaka. CCA security and trapdoor functions via key-dependent-message security. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 33–64. Springer, Heidelberg, August 2019. (Cited on page 2, 3, 15, 17, 18, 19, 20, 21)

[KNY21] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 31–61. Springer, Heidelberg, November 2021. (Cited on page 2)

[KW19] Venkata Koppula and Brent Waters. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 671–700. Springer, Heidelberg, August 2019. (Cited on page 2)

[LW21] Xu Liu and Mingqiang Wang. QCCA-secure generic key encapsulation mechanism with tighter security in the quantum random oracle model. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 3–26. Springer, Heidelberg, May 2021. (Cited on page 2, 21)

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 21–39. Springer, Heidelberg, August 2013. (Cited on page 3, 4, 29)

[Ms09] Steven Myers and abhi shelat. Bit encryption is complete. In *50th FOCS*, pages 607–616. IEEE Computer Society Press, October 2009. (Cited on page 2, 25, 29)

[MY22a] Tomoyuki Morimae and Takashi Yamakawa. One-wayness in quantum cryptography. Cryptology ePrint Archive, Report 2022/1336, 2022. `https://eprint.iacr.org/2022/1336`. (Cited on page 4)

[MY22b] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 269–295. Springer, Heidelberg, August 2022. (Cited on page 2)

[NC00] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. (Cited on page 6)

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990. (Cited on page 2)

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008. (Cited on page 2, 3, 4, 25, 29)

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990. (Cited on page 10)

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Heidelberg, August 1992. (Cited on page 2)

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, Heidelberg, March 2009. (Cited on page 2, 3, 4, 5, 25, 28, 29)

[RZ21] Bhaskar Roberts and Mark Zhandry. Franchised quantum money. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 549–574. Springer, Heidelberg, December 2021. (Cited on page 2)

[SGX23] Tianshu Shan, Jiangxia Ge, and Rui Xue. Qcca-secure generic transformations in the quantum random oracle model. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 36–64. Springer, 2023. (Cited on page 2)

[Sho98] Victor Shoup. Why chosen ciphertext security matters, 1998. IBM TJ Watson Research Center. (Cited on page 2)

[Unr14] Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 129–146. Springer, Heidelberg, May 2014. (Cited on page 6)

[Unr20] Dominique Unruh. Post-quantum verification of Fujisaki-Okamoto. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 321–352. Springer, Heidelberg, December 2020. (Cited on page 6)

[XY19] Keita Xagawa and Takashi Yamakawa. (Tightly) QCCA-secure key-encapsulation mechanism in the quantum random oracle model. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 249–268. Springer, Heidelberg, 2019. (Cited on page 2, 21)

[Zha12] Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012. (Cited on page 2)

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, August 2019. (Cited on page 6)